

# QL Today

Volume 16  
Issue 2  
Dez. - Feb.  
2011/12

ISSN 1432-5454

The Magazine about QL, QDOS,  
Sinclair Computers, SMSQ...

# Pimp my DVD

URS KÖNIG EXPLAINS  
IN DETAIL HOW TO  
IMPROVE THE  
QL TODAY DVD

In case the DVD becomes  
unreadable or physically  
damaged, please send an  
email to SMSQ@I-M-S.com

2011

[www.QLToday.com](http://www.QLToday.com)

# Contents

- 3 Editorial
- 4 News
- 8 Getting the table of contents of an Audio CD from the Internet under QPC - Part 2  
*Wolfgang Lenerz*
- 14 I2C Interface for QL Emulators - Part 3  
*Ian Burkinshaw*
- 18 Practising Scales *Geoff Wicks*
- 22 Making the most out of the DVD...  
*Urs König*
- 27 The DVD - Jochen's View *Jochen Merz*
- 28 E.V.A QL Review *P.Scott*
- 29 Programming in Assembler, Part 29  
Application Sub-Window Menus  
*Norman Dunbar*
- 41 SBASIC/SuperBASIC Reference Manual  
*Dilwyn Jones*
- 44 Last-Minute News
- 44 Small Ads

# Advertisers

in alphabetical order

Jochen Merz Software . . . . .	39
QLForum . . . . .	13
Quanta . . . . .	33
QuoVadis Design . . . . .	21

# QL Today

ISSN 1432-5454

## German office & Publisher:

Jochen Merz Software    Tel. +49 203 502011  
Kaiser-Wilhelm-Str. 302    Fax +49 203 502012  
47169 Duisburg    email: smsq@j-m-s.com  
Germany    email: QLToday@j-m-s.com

## Editor:

Geoff Wicks    Tel. +44 1332 271366  
Flat 5b    email: gtwicks@btinternet.com  
Wordsworth Avenue    email: QLToday@j-m-s.com  
Derby DE24 9HQ  
United Kingdom

## Co-Editor & UK Office:

Bruce Nicholls    Tel +44 20 71930539  
38 Derham Gardens    Fax +44 870 0568755  
Upminster    email: qltoday@q-v-d.demon.co.uk  
Essex RM14 3HA    email: QLToday@j-m-s.com  
United Kingdom

**QL Today** is published four times a year, our volume begins on beginning of June. Please contact the German or English office for current subscription rates or visit our homepage [www.QLTODAY.com](http://www.QLTODAY.com).

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email. We prefer ASCII, Quill or text87 format. Pictures may be in \_SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

**QL Today** reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is Copyright 2011 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Dilwyn Jones' web site where you find more information about lots of interesting QDOS software and INFOZIP at <http://www.dilwyn.uk6.net/arch/index.html>

**The deadline for the next issue  
is the 15th of February 2012!**

# Editorial

by Geoff Wicks

At first sight the QL is ending 2011 in a whimper. Certainly the QL-users email group has been deathly quiet for the last 2 or 3 months. But when I started to write the news pages I discovered that behind the scenes much activity is taking place.

Dilwyn Jones has devised an unusual way of solving a long standing software problem; Memory Lane Computing are continuing their hardware development; Norman Dunbar and George Gwilt have added to the qdosmsq wiki; the QL Forum are celebrating their 1st birthday; and Just Words is nearing completion of its maps project.

This year it is the new boys on the block who have been making the running.

Who had heard of Memory Lane Computing a year ago? Earlier this year they released a piece of hardware that was first suggested in 2007. The SER-USB is the first significant UK hardware development in 10 years.

A year ago two unknowns started the QL Forum. At the time several of us asked who they were. This year they are celebrating their first birthday with 100 members.

Quanta now has its first permanent magazine editor for over 6 years. He is a person who returned to the QL community just over a year ago.

This should give Quanta members food for thought. The six year rule, which requires committee members to step down after serving longer than six years on the committee, was designed to encourage new blood to invigorate the organisation. By recommending its continuance the committee are giving a clear message to the members, 'Don't expect us to be around for ever and ever, amen'

Quanta is in for a rough time next year. The present chairman is not eligible for re-election and QL Today knows of no 'heir apparent' waiting in the wings. In the worst case scenario Quanta will have to start winding up proceedings at the end of June 2012. When I met the Quanta committee last July I sketched out a constitutional scenario that would give Quanta a little extra breathing space to solve the problem. But that's all it would be, a little extra breathing space.

The Quanta subscription is rising by over 40%. I wonder how many members could name the year in which a chairman first warned that the subscription would soon have to rise. It was eight years ago at the 2003 AGM. Quanta is to be congratulated on holding off a rise for so long. Inevitably they have to be prepared for a sharp fall in membership, but those that remain will show that Quanta means something to them.

The other major change in Quanta is the proposed new constitution, heavily modelled on the present one. Some people, I know, are disappointed that the committee have not opted for more radical changes. To be honest I include myself among those people. But when I offered to do the work on the constitution I told the committee that at the drafting stage it was my job to be a servant and not a decision maker. I think we have produced a constitution that most members would want.

In the 28 years of its life the QL has faced many crises but has shown a resilience that has exceeded expectations. Quanta members should take heart from this. Whatever difficulties they may face next year they have the potential to survive, and to emerge from the problems stronger.

## Two Software Solutions?

Dilwyn Jones has released two programs that tackle a long standing QL problem described over the years as a 'chicken and egg' or a 'classic catch 22' situation.

When a file is saved under a QDOS environment a header is first created that contains various information about the file such as its length, file type and name. If the file is saved to a DOS environment the header is not recognised and thus not saved. QDOS can no longer recognise it as a valid QL file.

To get over this problem QL users have long used compression programs to transfer files to a DOS environment. This includes programs uploaded to QL websites. To decompress these programs QL-ers need an unzip program. Most QL-ers obtained their copy on floppy disk some years ago, but floppy disks are rapidly becoming obsolete and floppy drives have disappeared from many PCs. This poses a problem for people who have recently moved to software emulators from native hardware or who have recently returned to the QL scene. They need a copy of a QL unzip program but they can only use it after first unzipping it.

Jonathan Hudson has produced a self extracting version of QL unzip, but Dilwyn Jones reports that many users seek his help because they are unable to use this on some modern level 2 systems.

As a 'half way house' solution Dilwyn has written a basic program, Jobs2Bas, which converts a QL executable program into a basic program in which the executable programs details are stored in a series of data statements. As a basic program is a text file it can be copied to a DOS environment without becoming corrupt. The basic program can then be run to recreate the executable program.

Using Jobs2Bas Dilwyn has converted two versions of the QL unzip program to SuperBasic and these are downloadable from his website:

<http://www.dilwyn.me.uk/arch/index.html>

This enables QL users without floppy drives to download a QL unzip programme for their emulator.

In a similar development Malcom Lear has written a program to enable QPC users to read and write to DOS systems without the need for a QXL\_WIN file. In an email to Dilwyn Jones he wrote:

"Some time ago I worked on a system that stored QL headers in the file (appended at the end). It was developed to the point that my install of QPC ran all my programs and pointer environment straight from the PC drive without a QXL.win file. The problem of zipping was also solved."

This software is called QPConDOS and, as well as managing to store the executable file headers on a DOS/Windows drive, it patches the QL procedures and functions

EXEC, EXEC\_W, EX, EW, EXEP, EX\_M, ET, HOT\_LOAD, HOT\_CHP, HOT\_CHP1, HOT\_RES, and HOT\_RES1

to execute QL programs from a DOS/Windows directory.

Although the program is short, there is a lengthy procedure for setting it up which is fully explained in a installation document.

The program can be downloaded from Dilwyn Jones' website:

<http://www.dilwyn.me.uk/files/index.html>

## QL Web Activity

Although the QL scene has been fairly quiet this autumn, behind the scenes there has been some activity on QL websites.

Earlier this year QL Today reported that Dilwyn Jones had opened a Twitter account (@DilwynJones2) to announce changes to his website.

Adrian Ives of Memory Lane Computing has also opened a twitter account to enable users to keep in touch with product news and developments:

@memorylanecmptg

<http://twitter.com/#!/memorylanecmptg>

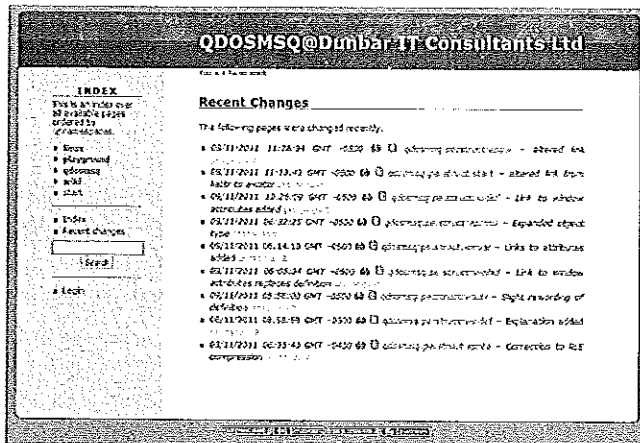
Memory Lane Computing can also be found on Facebook:

[www.facebook.com/memory.lane.computing](http://www.facebook.com/memory.lane.computing)

Norman Dunbar reports that there have been some major updates to his wiki on the internals of QDOS and the Pointer Environment.

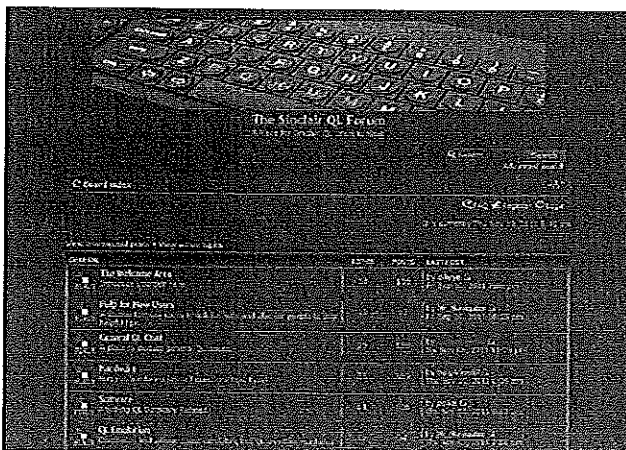
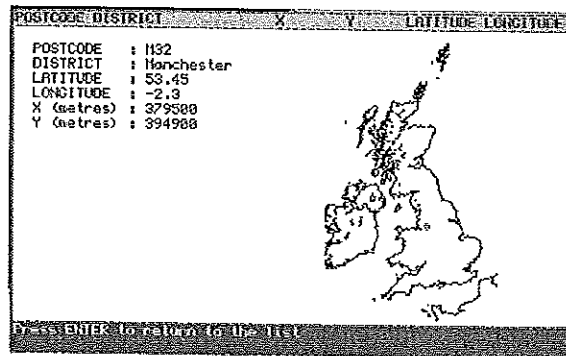
<http://qdosmsq.dunbar-it.co.uk>

In particular he praises the amount of effort George Gwilt has put into adding to, updating and correcting the information. As can be seen from the illustration, George made no fewer than 8 contributions in just two days.



Just Words! has tested the feasibility of the project by updating the postcodes program on its website using the UK maps database. This involved reducing the size of the database to one twentieth of its former size and then converting it into data lines. The quality that can be achieved can be seen from the illustration. Just Words! hopes to post the new version of the postcodes program on its website in the near future. At the moment there is no firm date for launching the QL maps page, but Just Words! hopes that it will be early in the new year. [www.gwicks.net/justwords.htm](http://www.gwicks.net/justwords.htm)

Peter Scott has reminded us that the QL Forum is now one year old. The site has become an established part of the QL scene. Some idea of its progress can be gained from the statistics. QL Today reported that in early February there were 367 posts on 84 different topics and 44 people had registered as members. In mid November the equivalent figures were 1457 posts on 246 topics with 100 people registered as members. [www.qlforum.co.uk](http://www.qlforum.co.uk)



## SOFTWARE NEWS

### Paragraph now Freeware

Dilwyn Jones writes: "The ProWesS-based pointer driven word processor Paragraph has now been released as Freeware by the author Francois Lanciault and added to the ProWesS page on my website. The version available to download is version 2.03. Paragraph was originally available in two versions, the original version 1 was available free and the enhanced version 2 was formerly available as commercial software – both versions are available from my website. Note that in order to use Paragraph you need to have both the pointer environment and ProWesS system on your computer. ProWesS is the PROGS Windowing System (hence the part capitalisation of the name ProWesS), originally developed by Joachim van der Auwera from the company PROGS in Belgium – ProWesS is also free to download from the same page these days."

Download Paragraph and ProWesS from <http://www.dilwyn.me.uk/prowess/index.html>

### Just Words! Mapping Progress

Just Words! reports that its mapping project is proceeding smoothly. All the present map databases have been converted to QL x,y coordinates to enable QL-ers to use them in their own programs. Just Words! intends to set up a second maps page for QL use consisting of the converted databases and a maps toolkit. This will include a detailed manual and several short SuperBasic programs:

- A simple map display program giving scaling advice
- A program to reduce the size of the databases
- A program to extract the data coordinates for a small area from a larger database
- A program to convert databases into SuperBasic data lines.

### Turbo Toolkit Upgrade

George Gwilt announces a further upgrade of the Turbo Toolkit:

"Version 3.39 allowed COMMAND\_LINE to select a daughter SBASIC's channel 0 by giving



as a parameter the ID of any channel opened to that SBASIC. The opened channel did not need to be a CON device.

Version 340 additionally allows the parameter to be the Job ID of the target SBASIC. COMMAND\_LINE looks first to see if the parameter is a Job ID."

The new version can be downloaded from <http://gwiltprogs.info/>

## Manuals Online

Dilwyn Jones has posted two manuals on his website:

### AURORA

"The manuals for the Aurora card are now available to download as PDF files from my website. Both the standard Aurora manual and the Aurora Technical Guide are available, thanks to a request from a QL user who stated a preference for using a PDF file reader to read QL manuals. The manuals are available from

<http://www.dilwyn.me.uk/docs/manuals/index.html> scroll down to the second half of the page, or go direct using <http://www.dilwyn.me.uk/docs/manuals/index.html#qubbesoft>

for this and other replacement manuals for products formerly sold by Qubbesoft".

### QIMI

"I have also added copies of manuals for the QIMI mouse interface. I have provided copies of both the earlier QIMI manual as sold by QJump and the slightly updated later version from Quanta. These are available as either Word.doc files or PDF files. These are in addition to the existing circuit diagram and register programming information, originally supplied by Dave Westbury." Download all these from

<http://www.dilwyn.me.uk/docs/manuals/index.html> and scroll down to the second half of the page, or go direct using

<http://www.dilwyn.me.uk/docs/manuals/index.html#qimi>

## HARDWARE NEWS

### Q\_BUS

Memory Lane Computing have announced a new project the Q-BUS. Adrian Ives describes it as:

"A programmable I/O device for use with standard QLs using the ROM port. The device will include a fast PIC32 chip to hide the complexity of attached storage devices so that only a very high level device driver will be required for

QDOS. It was originally going to form the basis of a new, higher speed, Ser-USB++ device, but I am looking at a better, cheaper, way of doing this now.

It is also intended that Q-BUS will be programmable with a new API, providing, for the first time, a standard way of accessing read/write peripherals over the ROM port.

The status of this project is still very much experimental, and it may never see the light of day as a commercial product. A lot depends upon how the software development goes and whether the performance will be good enough. The current software implements messaging between the QL and the PIC but not much else." Images can be found on the Memory Lane Computing website:

<http://www.memorylanecomputing.com/qbus.htm>

Adrian regularly posts update details of his projects on the QL-users email group. The latest news can be found on the website:

<http://www.memorylanecomputing.com>

Further information is contained in his blog.



Making the computers of yesterday relevant today

PRODUCTS SERVICES SHOP CONTACT BLOG LINKS

#### Q-BUS



Q-BUS is our new I/O device and is still in the prototype stage. It is a device that plugs into the ROM port of a standard QL and provides an I/O system based on a separate MCU (Micro Controller Unit). The MCU handles the interface with the external storage device, supporting a high level command structure to the QL so that any access structure is used by the QL device and can be returned to the Q-BUS. Preliminary documentation is here.

#### Inside the Q-BUS Prototype



#### Current Status

13/01/2011: Q-BUS hardware completed and tested. Work to date continues as the MCU that logic can be done.

Follow day to day progress with the prototype on our blog.



## Early AGM

Quanta has announced the dates for its 2012 AGM. It is to be held a month earlier than usual on the weekend of 24th and 25th March. The reason is that the chairman expects to be abroad on the usual date at the end of April.

The AGM will be important as it will occur in a year of change for Quanta. Next year subscriptions are due to rise by over 40%; a new chairman will have to be elected; and a proposed new constitution approved.

Quanta has announced some changes to the method of fixing subscriptions with the introduction of a Subscription comprising a Membership Fee and a Postal Surcharge. All members will pay a basic Membership Fee of £18. For those who opt for the electronic magazine, both UK and

verseas members, this will also be the Subscription. Members opting for the paper magazine will pay a Postal Surcharge which varies according to area of residence. The postal surcharge will rise each year in line with postage costs.

The present chairman has served on the committee since 2005 and has to step down next year under the six year rule. Should no one be prepared to replace her, then, under the constitution, she will remain in office for a further three months while efforts are made to find a new chairman. Should there still be no candidate then Quanta must start winding up proceedings at the end of June 2012.

The AGM will also vote on a revised constitution. Members received the draft of this in September and have until the end of the year to make further representations. The chairman reports in the latest Quanta Magazine that so far only one member has done so. The deadline of 31st December is important as the complexity of a complete revision of the constitution means that it cannot be approved clause by clause at the AGM.

### Can YOU do better?

Quanta has announced a competition for a new Quanta logo in the latest Quanta Magazine. The logo currently on the Quanta website is temporary. The editor of Quanta Magazine writes:

*"Any submissions must be web colour friendly, up to a maximum of 1000 x 1000 pixels, and supplied in two formats for bitmapped PNG and EPS."*

The competition will be judged by the committee at the AGM and there will be a small prize for the winner.



### QaLendar 2012

Dilwyn Jones writes: "My annual offering of the QaLendar (QL themed calendar) is now available to download from my website. Filled with QL themed pictures, it is available in two forms

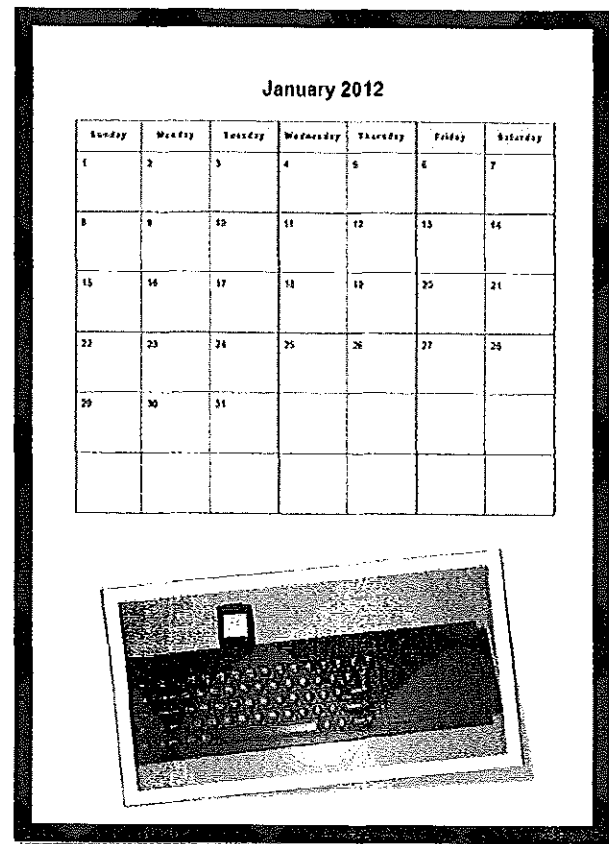
1. PDF file of the complete calendar with pictures. This is about 380KB to download.
2. The skeleton of the QaLendar as a 12 page Word .doc file, with just the one month per page and a placeholder 1x1 table at the bottom of each page in which you can add your own photos to customise it.

I chose not to add the Word .docx version of the QaLendar to the website as it is almost 10

MB in size and would take forever to upload and download.

Access it from the QaLendar page on my website at

<http://www.dilwyn.me.uk/gen/calendar/calendar.html>

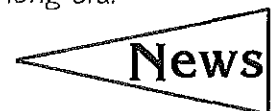


### Goodbye BBS?

At the end of November Tony Firshman announced that the main disk of his BBS had died.

*"It has been running pretty continuously on only two sets of computer hardware since the late 80s. It originally used a black QL with GC and Astracom modem. It is now in a boxed QL with Aurora, SGC, superHermes, Mplane and a 56k USR faxmodem. I used to link to Fidonet, and shared data with four or five QL BBSs (two or three in the UK, one in Holland and one in the USA) and the rest of the Fidonet community. Fidonet is long dead - killed by email and the internet. There are also no other QL based BBSs. Are there \*any\* dialup BBS systems still going?"*

*Does anyone use the BBS now? If not, it probably makes sense to retire this machine and use a modern multifunction laser printer. It will be the end of a very long era."*



# Getting the table of contents of an Audio CD from the Internet under QPC - Part 2

by Wolfgang Lenerz

## III - Getting the information from the internet

Connecting to an internet address from QPC is easy : just open a "tcp" channel - as can be seen in line 495:

```
ch%=FOP_IN('tcp_freedb.freedb.org:8880')
```

You now have opened a channel to port 8880 of freedb.org (8880 is the "port" on which freedb.org listens to requests/queries). Once you have

opened such a channel, you can just PRINT anything to that channel. This is then sent to the other computer on the internet (here one of the computers of freedb.org). You get the distant computer's reply via an INPUT made from the channel.

The entire dialogue with freedb is contained in the following procedure:

```
440 :
445 DEFINE PROCEDURE dialogue_with_freedb
450 REMARK chan% is a display channel
455 LOCAL lp%,a$,ch%,discid$,t% ,arr$(2,80),index% , artist$,track$
460 LOCAL longest%,time$,counter%,last%,temp% ,b$,title$
465 DIM current_returns$(100,100)
470 DIM subwinarray$(100,80) : REMARK on each query, a fresh array
475 subwinarray$(0)=0
480 arr$(0)=1
485 title$=""
490 debug%=main_lfl%(log_it%)
495 addprint "Starting to negotiate with freedb.org..."
500 ch%=FOP_IN('tcp_freedb.freedb.org:8880'):REMARK open channel to this internet address
505 IF ch%<0
510 lp%=warning%("No connection to the internet!",0,"Error!")
515 RETURN
520 END IF
525 IF inputt% ("200201","") <0
530 arr$(1)="Oops error in dialogue with server - 1"
535 add_to_array arr$
540 RETURN
545 END IF
550 :
555 addprint "OK - Sending hello to server..."
560 pprint "cddb hello wolfgang wolf.wolf.com smsqGet 001"
565 IF inputt% ("200","")<0
570 arr$(1)="Oops error in dialogue with server - 2"
575 add_to_array arr$
580 RETURN
585 END IF
590 :
595 addprint "Query DB for matching entries (Send disc ID)... : "
600 pprint make_disc_query_command$
605 lp%= inputt%("200211","211")
610 IF lp% <0
615 arr$(1)="Oops error in dialogue with server - 3"
620 add_to_array arr$
625 RETURN
630 END IF
635 :
```



```

640 IF lp%=211
645 REMark found inexact matches, so propose matches to user
650 counter%=current_returns$(0)
655 DIM arr$(counter%-2,76) : REMark number of inexact matches
660 FOR lp%=2 TO counter%
665 a$=getinfo$(2,current_returns$(lp%)) : REMark isolate string
670 t%= a$ INSTR current_returns$(lp%) : REMark find it
675 IF t%>0:arr$(lp%-2)=current_returns$(lp%,t% TO) : REMark get remainder of string
680 END FOR lp%
685 lp%=LIST_SELECT("Choose",arr$,,,,,,-1,-1)
690 IF lp%<0:RETURN
695 a$=current_returns$(lp%+2) : REMark the disc info
700 t%=" " INSTR a$
705 categ$a$(1 TO t%-1) : REMark global var
710 discid$=getinfo$(1,a$)
715 ELSE
720 a$=current_returns$(1) : REMark the disc info
725 categ$=getinfo$(1,a$) : REMark global var
730 discid$=getinfo$(2,a$)
735 END IF
740 addprint "Category : "&categ$
745 addprint "DiscID : "&discid$
750 pprint "cddb read "&categ$&" "&discid$
755 lp%=inputt$("210","210")
760 IF lp%>0
765 t%=current_returns$(0)
770 start%=0
775 longest%=0 : REMark longest artist name yet
780 counter%=toc(0,0) : REMark start of index into toc
785 longest%=24 : REMark name length
790 last%=-1 : REMark last title treated
795 DIM arr$(t%,DIMN (current_returns$,2)) : REMark make array as big as first one -
overkill
800 FOR lp%=2 TO t%
805 a$=current_returns$(lp%)
810 temp%= "TTITLE" INSTR a$ : REMark this could be a track title
815 IF temp% >0 AND temp%<LEN(a$)
820 t%=" " INSTR a$ : REMark find end of title tag
825 b$a$(temp%+6 TO t%-1) : REMark get title number
830 IF NOT is_number%(b$):NEXT lp%:EXIT lp% : REMark WHAT?????
835 temp%=b$ +1 : REMark title number
840 IF temp%>last%:last%=temp%
845 a$a$(t%+1 TO)
850 IF temp%<DIMN(arr$,1) :arr$(temp%)=arr$(temp%)&a$ : REMark make entry for it
855 ELSE
860 temp%="DTITLE" INSTR a$ : REMark is this the disc title?
865 IF temp%>0 AND temp%<LEN(a$)
870 temp%=" " INSTR a$
875 title$=title$&a$(temp%+1 TO) : REMark make title string
880 END IF
885 END IF
890 END FOR lp%
895 IF last%=-1
900 lp%=warning$("Number of tracks is too strange to handle",0,"Error!")
905 RETURN
910 END IF
915 arr$(0)=last%
920 FOR lp%=1 TO last%
925 a$=arr$(lp%)
930 temp%="/" INSTR a$
935 IF temp%=0:temp%=":" INSTR a$
940 IF temp%=0:temp%="-" INSTR a$
945 IF temp%>1 AND temp%<LEN(a$)-3
950 artist$=STRIP_SPACES$(a$(1 TO temp%-1))

```

```

955     track$=STRIP_SPACES$(a$(temp%+1 TO))
960     ELSE
965     artist$=a$
970     track$=""
975     END IF
980     IF lp%<DIMN(toc,1)
985     temp%=toc(lp%,4)
990     time$=temp% DIV 60
995     time$=make_length$(time$,2,"0",1)&":"
1000    temp%=temp% MOD 60
1005    time$=time$&make_length$(temp%,2,"0",1) : REMark isn't coercion wonderfful?
1010    ELSE
1015    time$=""
1020    END IF
1025    artist$=make_length$(artist$,art_length%," ",0)
1030    track$=make_length$(track$,track_length%," ",0)
1035    arr$(lp%)=artist$&" : "&track$&" "&time$
1040    END FOR lp%
1045    temp%="/" INSTR title$           : REMark make disc title/artist strings
1050    IF temp%=0:temp%=":" INSTR title$
1055    IF temp%=0:temp%="-" INSTR title$
1060    IF temp%<>0
1065    disc_artist$=STRIP_SPACES$(title$(1 TO temp%-1))
1070    disc_name$=STRIP_SPACES$(title$(temp%+1 TO))
1075    END IF
1080    add_to_array arr$
1085    END IF
1090    addprint "Sending quit command to server..."
1095    pprint "quit"
1100    lp%=inputt%("","")
1105    addprint 'Done'
1110    CLOSE#ch%
1115    make_main_appsub
1120    print_info
1125    FOR temp%=name_it% TO time_it%:main_lfl%(temp%)=1
1130    END DEFine dialogue_with_freedb
1135 :

```

In line 490 I set a variable to the state of the "log" item - if the log item is selected (which means that the corresponding menu flag "menu\_mfl%" is <math>\neq 0</math>), the entire dialogue is logged to the application subwindow. If not, only the result of the query is logged to the appsub window. Line 495 calls the "addprint" procedure : that just prints something to the log if the debug% variable is set (i.e. <math>\neq 0</math>). As a result, the "log" item controls whether everything, or only the result of the query, goes to the log.

Once I've opened a channel to freedb (line 500), I expect an answer from it: freedb notices that someone tries to connect to it and sends a greeting.

The input from freedb is always handled via the "inputt%" function, which I'll describe later. Suffice it to say here that the function checks some of the replies made by the freedb server and puts the actual reply data into a global variable called current\_returns\$, an array that is DIMmed in that function. If anything went wrong, the "inputt%"

function returns a negative error code, else the reply code from the server (more on which later). So, in line 525, I get the first reply from the freedb server. If this was an error, an early return is made. In lines 555 to 585 it's my turn to say hello to the server. The freedb server expects a string starting with "cddb hello " followed by a name, an internet address, the name of the program/protocol and a number, here "001". This is sent in line 560 and the reply handled via the "inputt%" function. The "pprint" procedure used, for example, in line 560 just prints the string to the tcp channel and, possibly, to the log.

Once these niceties are out of the way, it's time to query the freedb server for the disc in the drive and get its answer, so in line 600 I send it the disc query command. Again, the answer is put into the current\_returns\$ array by the "inputt%" function.

However, here, the freedb server can return two types of replies: either the disk is unique, and some information about the disc is returned, nota-

bly the category (rock, pop, classical etc) of the disc, and the "DiscID" which is the freedb internal number for the disc (nothing to do with the discid calculated from the disc itself). In this case, the reply sent back from the server contains the code "200". Or the disk isn't unique: it is possible that the discid calculated from the disc pertains to two different discs, which just happen to have the same number of tracks with the same lengths. In that case, the discid calculated will be the same for these discs. If that is the case, the server sends back a list of these discs and you will have to choose which one you want. In this case, the reply sent back from the server contains the code "211". So I can distinguish between the replies by checking the reply codes from the server.

This choice is handled by lines 640 to 710. In line 640, I check whether a multiple reply is made (code211) and act accordingly by getting the multiple disc names and proposing them to the user who then chooses one of them. Lines 720 and following handle the "unique" case.

In both cases I get the category and the DiscID from the reply. These two pieces of information are then needed for the main query, i.e. reading the information for the disc. The query for this is comprised of the literal string "cddb read" followed by the category and the DiscID and is sent to the server in line 750, the reply is obtained by line 755, again using the "inputt%" function.

The data sent back by the freedb server is contained in the current\_returns\$ array, put there by the "inputt%" function. Element 0 of that array contains the number of elements of that array actual-

ly containing data - this I put into the t% variable. Element 1 contains the name of the disc and the performing artist. Elements 2 to t% contain the info on each track. Now all that remains is to parse the data: Each track info (i.e. potentially each element of the array from element 2 onwards) contains the name of the artist and the name of the track, separated by " / ". If an element is a track info, the string will contain the literal string "TTITLE". If the element contains info about the disc, it will contain the string "DTITLE".

The loops in lines 800 to 890 and 920 to 1040 turn the data returned into nicely formatted strings (looking like three columns - formatted by the make\_length\$ function) which are then put into the application subwindow. The query to the freedb returns much information, not all about the tracks, and this loop filters out information about the tracks and the disc (using the TTITLE and DTITLE strings). Everything else is discarded. This is pretty much standard string slicing, splitting and concatenating, not much to talk about here. Finally, the result is added to the subwindow array "subwindowarray\$" (I told you I have great imagination for names) (line 1080) which is then used to fill in the application subwindow (line 1115).

Once out of the loop, I tell the freedb server that I'm done by sending it a "quit" string (line 1100) and, at the end, close the connection to the server, by closing the corresponding channel (line 1110). Finally, I show the new information in the window (lines 1115 and 1120).

That's about it. It only remains to dissect the "inputt%" function, which is as follows:

```

1135 :
1140 DEFine FuNction inputt% (expected_returns$,expect_several$)
1145 REMark this gets the return info from the internet freedb server & prints it
1150 REMark this returns negative value if error from server , else the server code
1155 REMark ch%= channel to internet
1160 LOCAL a$,b$,t%,lp%,index%
1165 INPUT#ch%,a$ : REMark get reply from server
1170 IF LEN (a$)>2
1175 IF a$(LEN(a$))==CHR$(13): a%=a$(1 TO LEN(a$)-1) : REMark strip trailing newline
1180 END IF
1185 index%=1
1190 DIM current_returns$(100,100)
1195 current_returns$(index%)=a$
1200 current_returns$(0)=1
1205 addprint " <--- "&a$
1210 IF expected_returns$="":RETURN 0 : REMark all OK since no expected return
1215 t%=" " INSTR a$
1220 IF NOT t%:RETURN -1 : REMark what, no code back from server?
1225 b%=a$( TO t%-1)
1230 IF LEN(b$)<>3 : RETURN -1 : REMark oops, return is three digits code
1235 t%=b$ INSTR expected_returns$

```

```

1240 IF t%=0:RETurn -1 : REMark oops, error code
1245 IF t% MOD 3<> 1: RETurn -1
1250 IF is_number% (b$)
1255 t%=b$
1260 ELSE
1265 RETurn -1
1270 END IF
1275 IF NOT b$=expect_several$:RETurn t%
1280 REPEAT lp%
1285 INPUT#ch%,a$
1290 IF a$="":EXIT lp%
1295 IF a$(1)="":EXIT lp%
1300 IF LEN (a$)>2
1305 IF a$(LEN(a$))=CHR$(13): a%=a$(1 TO LEN(a$)-1): REMark strip trailing newline
1310 END IF
1315 addprint " "&a$
1320 index%=index%+1
1325 current_returns$(index%)=a$
1330 END REPEAT lp%
1335 current_returns$(0)=index% : REMark last element used
1340 RETurn t%
1345 END DEFine inputt%
1350 :

```

This function returns either a negative error code if something went wrong, or the answer code returned by freedb: freedb generally prepends a code (followed by a space) to its answers. The code is a three digit number - numbers 200 and 201 generally mean that the query was successful. The "inputt%" function takes two parameters: first all of the expected return codes from the freedb server, and then a return code indicating that there should be several lines of text as answer, and not only just one line. The expected return codes from the server are checked against the return code actually returned by the server - if they don't match, then the function returns an error code.

You will note that the reply from the freedb server is obtained via the INPUT statement (at line 1065.) Again, this is pretty dangerous: I don't know what happens in case of a communications breakdown. Potentially, the input statement might then hang forever as it doesn't get a complete line of text. This has NOT happened to me, but could happen to you, so this could be another area for improvement. (This is one area where I have always felt S\*Basic to be sub-optimal: an INPUT statement with a timeout would have been so nice..).

Anyway, the line returned from the server gets a trailing PC newline stripped off (line 1175), the global array variable "current\_returns\$" is DIMmed and the return from the server is (maybe) added to the log (line 1205). Then the return code from the server is checked against the expected returns and an error code is returned if this

doesn't match (lines 1225 to 1270). The function returns early if only one line is expected (line 1275), if not it obtains the other lines in the loop in lines 1280 to 1330. Again, the reply is obtained with the INPUT statement, so the same comments on that apply. That loop exits if I get an empty line from the server (which shouldn't happen) or if it get a line starting with a decimal point - the signal from the server that it has finished replying. The replies from the server are put into the current\_returns\$ array (line 1325). Warning : no check is made whether this wouldn't overshoot the end of the array. The array is DIMmed to (100,100) at line 1190. That **\*\*should\*\*** be sufficient for any reply from the server (I'm not aware of a CD with more than 100 tracks) but one never knows! At least, if this happens, the program will just stop with an error and not cause any overflows... Finally, the function fills in element 0 of the current\_returns\$ array, with the number of elements actually used (line 1335). This will later avoid me having to check against empty elements - I'll know exactly how many elements I have to deal with.

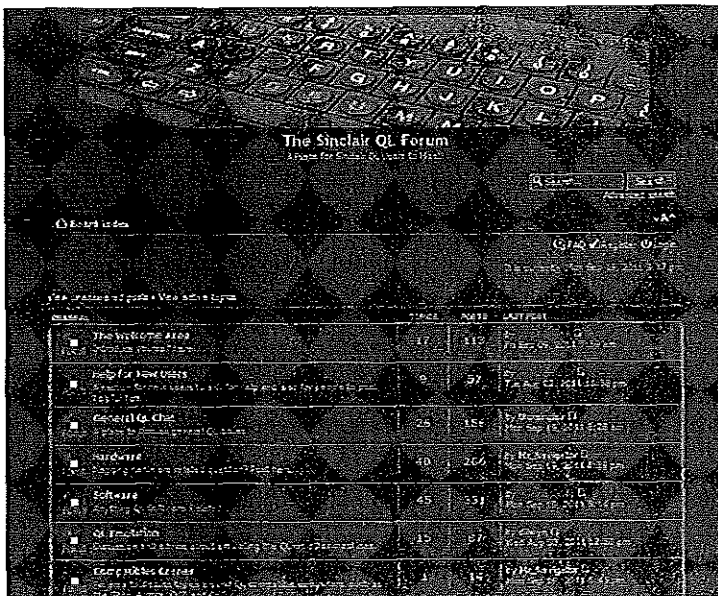
That's more or less it. The program is NOT well error checked, much could go wrong (but hasn't here, yet) - it's more a proof of concept than anything else.

I hope this was interesting, at least to some. I did show that connecting to the internet from a QL platform IS possible. I think, even, that a simple mail program using a POP server might be possible - but I don't see myself writing that! Keep on QLing.

<http://qlforum.co.uk>



**A Place for Sinclair QL Users to Meet!**



**THE QL FORUM** is an informal online community for talking about all things QL.

**Join in and get involved!**

# I2C Interface for QL Emulators

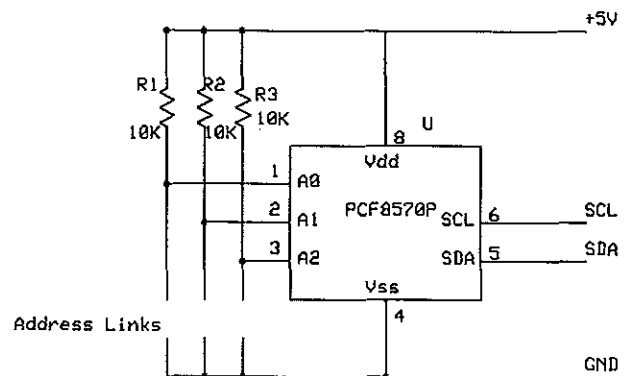
## Part 3

by Ian Burkinshaw

In part one of this series we looked at some of the basics of using the I2C bus and the ByVac BV4221 USB to I2C converter. In the first part I also covered the I2C interface using the PCF8574 parallel device. In part two we looked at using the PCF8591 analogue to digital (DA) and digital analogue converter device and the DS1307 RTC (Real Time Clock). This time we will look at the PCF8570 256 x 8 RAM and the DS1803 Dual digital potentiometer.

As can be seen from the circuit below, the PCF8570 RAM is very simple. Just power, GND and the SCL and SDA I2C bus lines not forgetting the address pull up resistors and links if required. As we have seen before address links are only required if more than one device of this type is required in your application. The PCF8570 is a 256 x 8 bit RAM (Random Access Memory) - put another way it stores 256, 8 bit words. Now that may not seem much these days, but it can have its uses. For example, you may wish to store some variables while being able to reset your QPC session, then load those variables back again. The PCF8570 is volatile memory, that means it will forget every thing stored if you remove the power from the device. However if

the device is powered with a battery or an alternative power supply when the QPC/QL is turned off, it would retain all the data. The device has a low standby current of 15uA, so a battery could last for years. As you can see from the circuit diagram the PCF8570 has address links, please see part one of this series for the address ranges.



The short test example program below tests each of the 256 memory locations. The REM statements explain what is going on in the program.

```

10 REMark RTC (Real Time Clock) DS1307 test routines
20 init
40 OPEN#3;ser2ir:REMark i=ignor hardware handshake, r=raw data
50 PRINT#3;CHR$(13);:REMark Carriage Return to set the baud rate in the USB to I2C
  converter, required on first pass to initialise USB to I2C converter.
60 print_reply:PRINT "Reply from sending CR."
70 PRINT
80 PRINT#3;"V";CHR$(13);:REMark Command to USB to I2C coverter for firmware version.
90 PRINT "Return USB Converter Version Number:-";
100 extract_read_data:PRINT d$;print_reply:REMark Prints version number reply from USB to
I2C converter
110 PRINT
120 PRINT#3;"D";CHR$(13);:REMark Sets USB to I2C coverter to receive decimal numbers,
  default is hex numbers.
130 PRINT "Decimal Mode Selected"
140 print_reply:REMark returns a device address.
150 PRINT
180 PRINT "Writing RAM Area Data"
190 write_ram_data
200 PRINT "RAM Data read from device"
210 read_ram_data
220 PRINT "From the first data byte the number should match between what was written to the
  device and what is read from the device."
280 AT 10,10:PRINT "End          ":CLOSE#3:STOP
290 :
1000 DEFine PROCedure init
  
```



```

1010 CLS
1020 BAUD 115200
1030 ram=174:REMark PCF8570 address, all address links open.
1040 parallel1=126:REMark PCF8574A address, all address links open
1050 parallel2=78:REMark PCF8574 address, all links open
1060 adda=158:REMark PCF8591 address, all address links open
1070 rtc=208:REMark DS1307 real time clock, one fixed address with this device.
1080 digpot=94:REMark DS1803 Digital Poteniometer, all link open
1090 DIM tdata(7)
1100 DIM days$(7,3)
1110 RESTORE
1120 FOR a=1 TO 7
1130 READ d$
1140 days$(a)=d$
1150 NEXT a
1160 DATA "Mon","Tue","Wed","Thu","Fri","Sat","Sun"
1170 END DEFine init
1180 :
1190 DEFine PROCedure print_reply
1200 c$=""
1210 REPeat loop
1220 a$=INKEY$(#3)
1230 b$=a$
1240 c$=c$&b$
1250 PRINT b$;
1260 IF a$="," THEN EXIT loop
1270 END REPeat loop
1280 END DEFine print_reply
1300 :
1310 DEFine PROCedure non_print_reply
1320 c$=""
1330 REPeat loop
1340 a$=INKEY$(#3)
1350 b$=a$
1360 c$=c$&b$
1370 IF a$="," THEN EXIT loop
1380 END REPeat loop
1390 END DEFine non_print_reply
1400 :
1410 DEFine PROCedure extract_read_data
1420 d$=""
1430 REPeat data_loop
1440 a$=INKEY$(#3)
1450 b$=a$
1460 d$=d$&b$
1470 IF a$=CHR$(10) THEN EXIT data_loop
1480 END REPeat data_loop
1490 END DEFine extract_read_data
2110 :
2120 DEFine PROCedure write_ram_data
2140 FOR ramd=0 TO 255
2150 dd=RND(0 TO 255):REMark Generating a random number to load into the RAM
2160 PRINT#3;"s-";ram;" ";ramd;" ";dd;" p";CHR$(13);:REMark the first number after the
s-ram is the starting word address, the remaining numbers are data to be loaded
incrementing the word address for each data item sent.
2170 non_print_reply
2180 PRINT dd;" ";
2190 NEXT ramd
2200 PRINT
2210 END DEFine write_ram_data
2220 :
3000 DEFine PROCedure read_ram_data
3010 FOR a=0 TO 255
3020 PRINT#3;"s-";ram;" ";a;" p";CHR$(13);:REMark the first number after the s-174 is the
starting word address, when reading data this set the start word address.
3030 non_print_reply
3040 PRINT#3;"s-";ram+1;" g-1 p";CHR$(13);:REMark g-9 means it will read 9 data words in
this example.

```

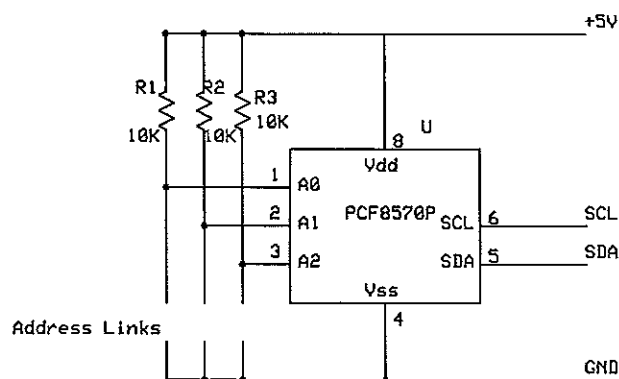
```

3050 extract_read_data:non_print_reply
3060 d=d$
3070 PRINT d;" ";
3110 NEXT a
3120 PRINT
3130 END DEFine read_ram_data
3140 :

```

The final device we will look at in the series is the DS1803 dual potentiometer. This is a very interesting device. It has two full independently controlled via I2C interface potentiometers. There are 3 versions of this device, DS1803-10 which the potentiometers have the value 10K ohms, DS1803-50 which is 50K ohms and DS1803-100 which is 100K ohms. End potentiometer can be incremented in 256 steps, each step increasing the resistance of the potentiometer by the same amount, so is linear in operation. Making a logarithmic potentiometer is done in your software code, however to achieve this the number of control steps will reduce. For an audio application I was working on I found 26 steps worked quiet well. One thing you must ensure is that any voltage applied to the potentiometer connections must not exceed the range 0V to +5V from the device ground. There is not any electrical isolation between the control logic, I2C bus or the potentiometer elements with in the device hence the caution required. As can be seen from the

circuit the DS1803 is very simple. Just power, GND and the SCL and SDA I2C bus lines not forgetting the address pull up resistors and links if required. As we have seen before address links are only required if more than one device of this type is required in your application. As you can see, from the circuit diagram the DS1803 has address links, please see part one of this series for the address ranges. The software listing assumes the links are open.



```

10 REMark Digital Poteniometer DS1803 test routines
20 init
40 OPEN#3;ser2ir:REMark i=ignor hardware handshake, r=raw data
50 PRINT#3;CHR$(13);:REMark Carriage Return to set the baud rate in the USB to I2C
  converter, required on first pass to initialise USB to I2C converter.
60 print_reply:PRINT "Reply from sending CR."
70 PRINT
80 PRINT#3;"V";CHR$(13);:REMark Command to USB to I2C coverter for firmware version.
90 PRINT "Return USB Converter Version Number:-";
100 extract_read_data:PRINT d$:print_reply:REMark Prints version number reply from USB to
  I2C coverter
110 PRINT
120 PRINT#3;"D";CHR$(13);:REMark Sets USB to I2C coverter to receive decimal numbers,
  default is hex numbers.
130 PRINT "Decimal Mode Selected"
140 print_reply:REMark returns a device address.
150 PRINT
160 potval0=128
170 potval1=255
180 PRINT "Writing Pot 0 Data"
190 write_pot_0_data
200 PRINT "Reading Pot 0 Data"
210 read_pot_0_data
220 PRINT "Writing Pot 1 Data"
230 write_pot_1_data
240 PRINT "Reading Pot 1 Data"
250 read_pot_1_data
280 PRINT "End          ":CLOSE#3:STOP
290 :
1000 DEFine PROCedure init
1010 CLS
1020 BAUD 115200

```

```

1030 ram=174:REMark PCF8570 address, all address links open.
1040 parallel1=126:REMark PCF8574A address, all address links open
1050 parallel2=78:REMark PCF8574 address, all links open
1060 adda=158:REMark PCF8591 address, all address links open
1070 rtc=208:REMark DS1307 real time clock, one fixed address with this device.
1080 digpot=94:REMark DS1803 Digital Poteniometer, all link open
1090 DIM tdata(7)
1100 DIM days$(7,3)
1110 RESTORE
1120 FOR a=1 TO 7
1130 READ d$
1140 days$(a)=d$
1150 NEXT a
1160 DATA "Mon","Tue","Wed","Thu","Fri","Sat","Sun"
1170 END DEFine init
1180 :
1190 DEFine PROCEDURE print_reply
1200 c$=""
1210 REPEAT loop
1220 a$=INKEY$(#3)
1230 b$=a$
1240 c$=c$&b$
1250 PRINT b$;
1260 IF a$="," THEN EXIT loop
1270 END REPEAT loop
1280 END DEFine print_reply
1300 :
1310 DEFine PROCEDURE non_print_reply
1320 c$=""
1330 REPEAT loop
1340 a$=INKEY$(#3)
1350 b$=a$
1360 c$=c$&b$
1370 IF a$="," THEN EXIT loop
1380 END REPEAT loop
1390 END DEFine non_print_reply
1400 :
1410 DEFine PROCEDURE extract_read_data
1420 d$=""
1430 REPEAT data_loop
1440 a$=INKEY$(#3)
1450 b$=a$
1460 d$=d$&b$
1470 IF a$=CHR$(10) THEN EXIT data_loop
1480 END REPEAT data_loop
1490 END DEFine extract_read_data
2000 :
2010 DEFine PROCEDURE write_pot_0_data
2020 PRINT#3;"s-";digpot;" ";169;" ";potval0;" p";CHR$(13);
2030 non_print_reply
2040 END DEFine write_pot_0_data
2050 :
2060 DEFine PROCEDURE write_pot_1_data
2070 PRINT#3;"s-";digpot;" ";170;" ";potval1;" p";CHR$(13);
2080 non_print_reply
2090 END DEFine write_pot_1_data
2100 :
3000 DEFine PROCEDURE read_pot_0_data
3010 PRINT#3;"s-";digpot+1;" g-1 p";CHR$(13);
3020 extract_read_data:non_print_reply
3030 d=d$
3040 PRINT d;" ";
3050 PRINT
3060 END DEFine read_pot_0_data
3070 :
3080 DEFine PROCEDURE read_pot_1_data
3090 PRINT#3;"s-";digpot+1;" g-2 p";CHR$(13);
3100 extract_read_data:non_print_reply

```

```

3110 d=d$
3120 PRINT d$;" First number is the value for pot 0 the second number is for pot 1"
3130 PRINT
3140 END DEFine read_pot_1_data
3150 :

```

That concludes our quick overview of some devices that can be controlled using the I2C bus and how to use the I2C bus. Next time we start to look at some applications, first being, driving LCD alpha numeric displays.

## References

PCF8570 Ram Data Sheet  
[http://www.nxp.com/documents/data\\_sheet/PCF8570.pdf](http://www.nxp.com/documents/data_sheet/PCF8570.pdf)  
 DS1803 Digital Potentiometer Data Sheet  
<http://datasheets.maxim-ic.com/en/ds/DS1803.pdf>

# Practising Scales

by Geoff Wicks

I wonder how many of us have fully mastered the SCALE keyword. It is one of the more tricky SuperBasic commands to use with many traps for the unwary. Recently I have been looking at the keyword more systematically than I have done previously. As part of the Just Words! mapping project I had to write a simple map viewing program. As each map has a different size and a different geographic location, each map requires its own custom scaling. The program had to work out the optimum scaling in each case.

There is no better way to learn than by practical work. This article is an exercise in which we are going to learn from mistakes by deliberately falling into some of the scaling traps. Your machine should be fired up and ready to go.

First let's remind ourselves of the syntax of the SCALE keyword:

```
SCALE [#channel,]height,coordinates of
bottom left hand corner
```

First, we need to have our machine in TV mode 4 resolution. For most of us a simple matter of the Toolkit II command:

```
WTV 4
```

Now type in:

```
PAPER 7 : INK 0 : CLS
```

We are not going to type in a program, but instead a number of commands. Thus we have a main window and an input window.

We type in our first graphics command:

```
CIRCLE 0,0,10
```

We notice straight away that only a quarter of the circle appears on screen. Most of us know

the QL can print outside the visible screen, but it is surprisingly easy to forget this when you are developing a program. If nothing appears on the screen we often embark on a fruitless search for an error in our code when we really need to get the scaling correct. As we shall see the problem is often with the x-coordinate.

Let's print a circle in the centre of the screen. We know that WINDOW #1 in TV mode is 448 x 200 and thus the centre of the screen is at 224,100. Enter

```
CIRCLE 224,100,10
```

No circle is to be seen. This means it is being printed outside the visible screen area, and thus there is a scaling problem.

When you open a window without defining a scale it defaults to SCALE 100,0,0. This means the window is 100 graphics units high and has an origin at 0,0. Thus for graphics the height of our window is not 200 screen pixels but 100 graphics units. Try:

```
CIRCLE 50,50,10
```

This time we see the circle is centred vertically, but not horizontally. It appears to the left of centre. If a window is 200 pixels high and that is equivalent to 100 graphics units, then a width of 448 pixels it is equivalent to 224 graphics units. Try:

```
CIRCLE 112,50,10
```

Once again the circle is centred vertically, but not horizontally. This time the circle appears to the right of the centre. In other words the x,y ratio of graphics units is not the same as the x,y ratio of the screen pixels. This is one of the things that makes the SCALE keyword difficult to use.

If a window has 100 vertical graphics units, then how many horizontal graphics units does it have?

In her book 'QL SuperBASIC' Jan Jones neatly skirts around this problem with the words, 'the number of units wide that the window is is deduced from the height'. Jan Jones does not tell us how to do this but later gives an example of a screen of 137 x 100 pixels as being a square for graphics purposes. We can thus deduce that the x:y ratio of a graphics screen is 1.37:1.

Thus if a 448 x 200 window has 100 vertical graphics units, then it has  $224 / 1.37 = 163.5$  horizontal graphics units. If you want to check this on our window try:

```
POINT 163,50
```

You will see a small dot centred vertically at the far right of the screen.

So let's try:

```
CIRCLE 82,50,10
```

This time we have a circle at the centre of the screen.

It may seem strange to have a graphics ratio different from the pixel ratio but the advantage of this system is that usually your graphics will be correctly displayed on screens of different size.

You will have noticed by now that although we are using the CIRCLE keyword, we are not getting a true circle on the screen. If you want something that looks like a circle try:

```
ELLIPSE 82,50,10,1.37,PI/2
```

Can you deduce why this works?

Now a final bit of practical work. This time we are going to change the scaling to give a larger graphic and will move the origin of the graphic to the centre of the screen. We are going to make the height 20 graphics units and this means the graphics y-coordinate at the centre of the screen will be 10. The number of vertical graphics units is 1/10 of the number of pixels and thus the width of the screen will be  $448 / (10 \times 1.37) = 32.7$ . Thus the graphics x-coordinate at the centre of the screen will be 16 to the nearest whole number. Now type in:

```
CLS
```

```
SCALE 20,-16,-10
```

(Don't forget that the co-ordinates are the position of the bottom left hand corner of the screen!)

Now draw our circle:

```
CIRCLE 0,0,5
```

and

```
CIRCLE 0,0,6
```

Now try

```
CIRCLE 0,0,5.5
```

And for good measure

```
CIRCLE 0,0,5.25
```

```
CIRCLE 0,0,5.75
```

When drawing graphics you do not have to stick to whole numbers. It would be impossible to draw high quality maps if you could only use whole numbers. The first 10 coordinates of the British Isles map database are:

1.906,59.625

1.908,59.621

1.910,59.617

1.911,59.612

1.913,59.607

1.914,59.602

1.916,59.597

1.917,59.591

1.917,59.585

1.918,59.579

How do we determine the scaling to be used in a graphic? The purists tell us we should have everything worked out on paper before we start to program, but I suspect that is not the way most of us want to work.

My suggestion is to initially develop your graphic on a full size screen even if you only want it displayed on a part of the screen. This makes it easier to determine the scaling, which you can then resize and move at your leisure.

How did I automate the scaling on my map display program? To determine the height I first looked at the maximum and minimum y-coordinates and calculated the range. I then did this for the x-coordinates. I made my height a little greater than the larger of the two.

Having done this it was fairly easy to determine the y-coordinate. The centre of the screen is half of the height and I also knew the range of the y-coordinates. The calculation was simple:

y-coordinate = (height - yrange)/2 rounded off to a whole number.

The x-coordinate was more difficult because the width of the screen in graphics units varies according to the width of the screen in pixels:

width in graphics units = (1.37 x height x pixel width)/pixel height

Then

x-coordinate = (width - xrange)/2 rounded off to a whole number.

If you want to rescale your graphic there are three things to remember:

- 1: The larger the height, the smaller the graphic.
- 2: Increasing the x-coordinate moves the graphic to the right, decreasing it to the left.
- 3: Increasing the y-coordinate moves the graphic upwards, decreasing it downwards.

Finally let's look at the problem that led me to study the SCALE keyword in some detail.

Imagine your atlas open at a map of the world. That map has to become a QL graphic using longitude and latitude as the basis for x and y co-ordinates. The origin of the graphic is at longitude 0, latitude 0, a point in the Atlantic Ocean off Gabon in Africa.

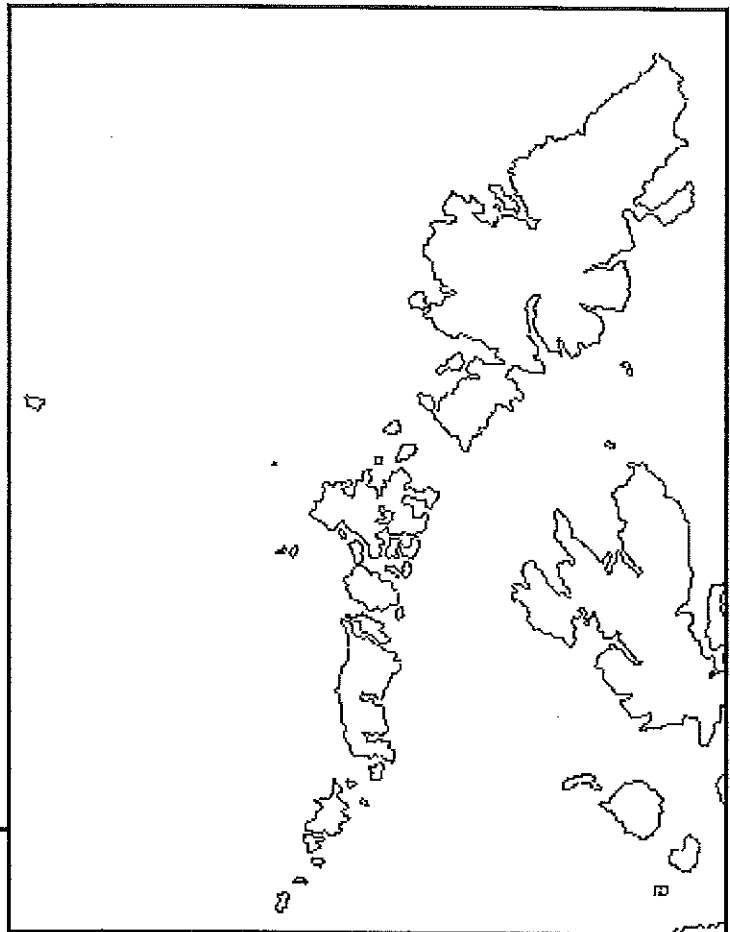
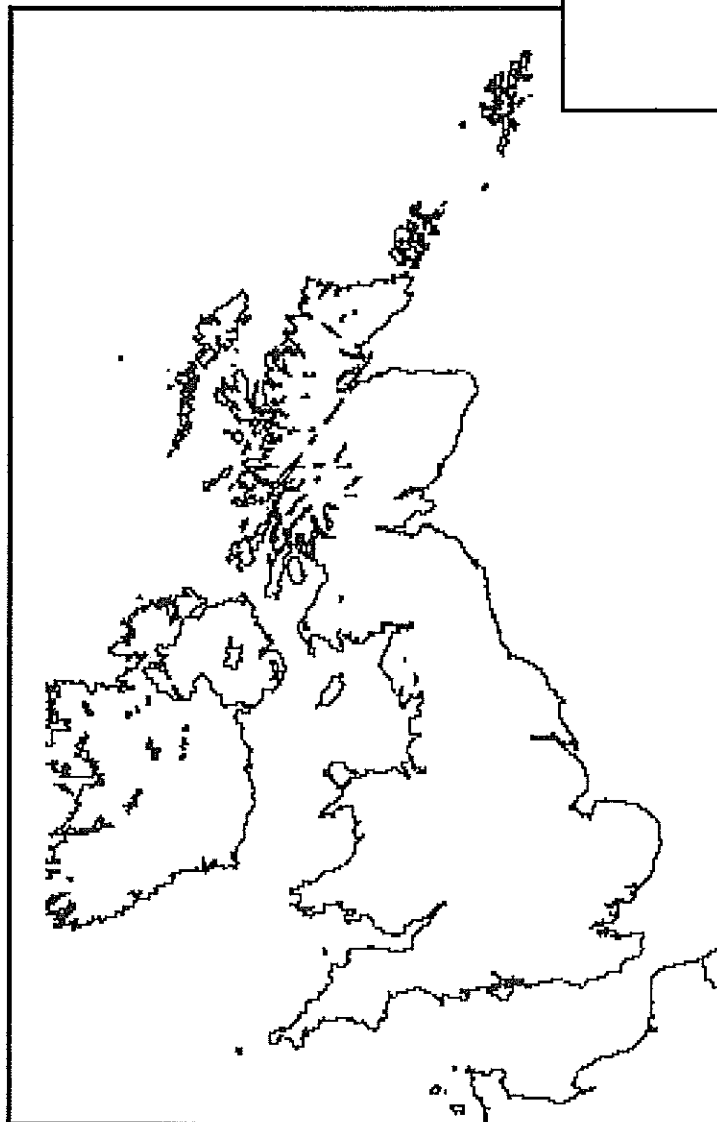


Figure 2 (above): SCALE 4,-8,69



The British Isles are just a small part of the graphic but using the SCALE command we can print out a detailed image of it as can be seen in figure 1.

But we can go a good deal further. Off the west coast of Scotland are the Hebridean islands and we can even draw a detailed map of those. On the far left of figure 2 is a small island – part of the St. Kilda archipelago – the most remote part of the British Isles. This island is about four and a half kilometres from North to South and three and a quarter kilometres from East to West. Its shape not is reproduced accurately, but it is still visible. And that is on a graphic of the whole world.

The SCALE keyword is truly powerful.

Figure 1 (left): SCALE 25,-20,54



QUO VADIS  
DESIGN

Independent Information  
Technology Services

www.ql-qvd.com

QL/QDOS/SMSQ/E Software

QUO VADIS  
DESIGN Independent Information  
Technology Services

QL/QDOS/SMSQ/E Software

[Home](#) [Products](#) [Support](#) [Company](#) [Contact](#)

#### Welcome

Quo Vadis Design sells software for the Sinclair Quantum Leap computer (QL) and variants including a new OS called SMSQ/E.

The QL is a computer in its 25th year Anniversary.

The Sinclair QL - a quantum leap in personal computing



Software emulations of the QL now exist which can run on a PC/Mac with Windows/Linux or Mac Operating systems.

#### News

★ [QVD QL News Blog - keep up to date](#)

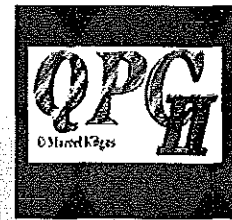
[News Blog](#)

24/02/2009

★ [Quo Vadis Design Website Launched](#)

01/02/2009

#### FEATURED PRODUCT



BUY NOW!

Copyright © 2009 Quo Vadis Design. All Rights Reserved.

[Home](#) | [Products](#) | [Support](#) | [Company](#) | [Contact](#) | [Privacy](#)

[Bruce@ql-qvd.com](mailto:Bruce@ql-qvd.com)

Quo Vadis Design  
38 Derham Gardens  
Upminster  
RM14 3HA  
UK

Tel: +44 (0)20 71930539  
Fax: +44 (0)870 0568755

Special Offers available from  
Jochen Merz Software for its  
25 years in QL Trading

Check the QL News Blog on  
our website for the special  
offers



Subscriptions taken online

# Making the most out of the DVD...

by Urs König

## Before I start

With the last issue of QL Today came "The DVD 2011". Page 36 gave you all the information needed to use this DVD, but if you want to get more out of the DVD this article is for you.

## What isn't it?

Please keep in mind that the DVD is made for use with a computer and not a DVD player. It does not have any VIDEOS/MOVIES on it. Neither does it have any AUDIO, well at least not what non-tinkerers will expect from a regular DVD/CD. The only "things" you can hear from are

- 1) the SOUND of a Cash Register in the "Simple HTML Tutorial" by Norman Dunbar (location `\QLrepository\doc\htmltut\`),
  - 2) all the BEEPs the virtual QLs like QPC2 or Q-emulator will generate and
  - 3) all the SOUNDS the "Sampled Sound System (SSS)" of QPC2, Q-emulator or Q40/Q60 will play.
- That's it. No more, no less.

## What is it then?

So what is the DVD good for? It has a lot of material for you to explore. To be precise the DVD holds 26,014 files in 1,842 folders. Altogether there are 3.49 GBytes of documents, software, pictures and other data. The DVD has a HTML top level menu (`index.htm`) and is layed out so that this menu starts automatically on systems configured to allow auto-start of DVDs. The menu itself is considered to be self-explanatory, lists all the main contents, lets you open every issue of the scanned QL Today magazines (that was the initial motivation to produce this DVD), enter every single "QL (TOPIC) collection"<sup>1</sup>, allows you to open the "README first" document of each collection and finally access some websites. Occasional users shall be happy with the DVD as it is. Regular users and hardcore QLers will probably copy the DVDs content to their computer(s) or even to their file server(s) and will store the physical DVD away.

## The content

The scanned QL Today magazines are located in the `\PDFs\` folder. The various collections are as follows:

TOPIC	Physical location	QXL.WIN
Local (offline) copy of Dilwyn Jones's QL website	<code>\QLrepository\djw\</code>	-
QL DOCUMENTATION collection v2.93	<code>\QLrepository\doc\</code>	X
QL HISTORY, 25TH ANNIVERSARY edition	<code>\QLrepository\25a\</code>	-
QL EMULATORS collection v1.30	<code>\QLrepository\emu\</code>	X
QL ON A STICK environment	<code>\QLrepository\qos\</code>	X
QL TOOLKITS collection v1.01	<code>\QLrepository\tks\</code>	X
QL GAMES collection v1.01	<code>\QLrepository\gms\</code>	X
QL PICTURE GALLERY collection v3.00	<code>\QLrepository\pic\</code>	-

## Strip it down first

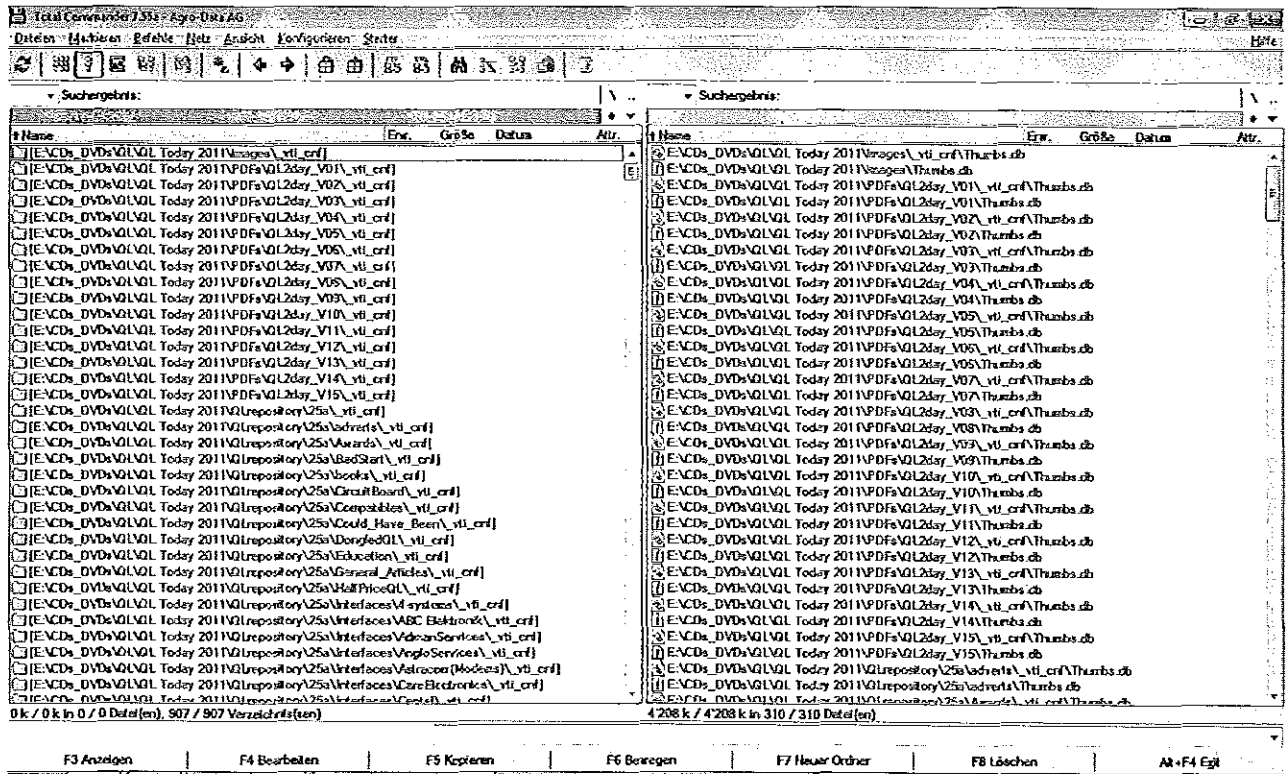
For some a 1:1 copy of all the files and folders of the DVD may be just fine, for others, including me, it could be called "sub-optimal". I use the DVD's content as I do any other QL related material on my PCs, where I rely on local search engines (software) and picture galleries (software). Working that way I discovered there are many files on the DVD which are of no use to me. So I decided to strip the DVD down.

<sup>1</sup> Such TOPICs are each QL CD maintained by Dilwyn Jones for over a decade now and the QL PICTURE GALLERY produced by the author.

Here's what I've deleted:

- All `_vti_cnf` folders, resulting in 13005 files and 907 folders (5.07 MBytes) deleted. The `"_vti_cnf"` stuff are artefacts which have been automatically created by Microsoft's FrontPage software package while Jochen edited the DVDs content.
- All `Thumbs.db` files, resulting in 154 files (4.07 MBytes) deleted. `"Thumbs.db"`, produced by the Windows XP operating system for its own need (to speed up the explorer image preview), are binary files holding thumbnail images of the pictures in that folder. Your computer will decide if and how thumbnails are processed, so you don't need those pre-generated files from the DVD.
- Selective duplicate files between `"QL (TOPIC) collections"` have been deleted, but only when those files are considered off-topic for the specific collection or are located in wrong or multiple places without good purpose.

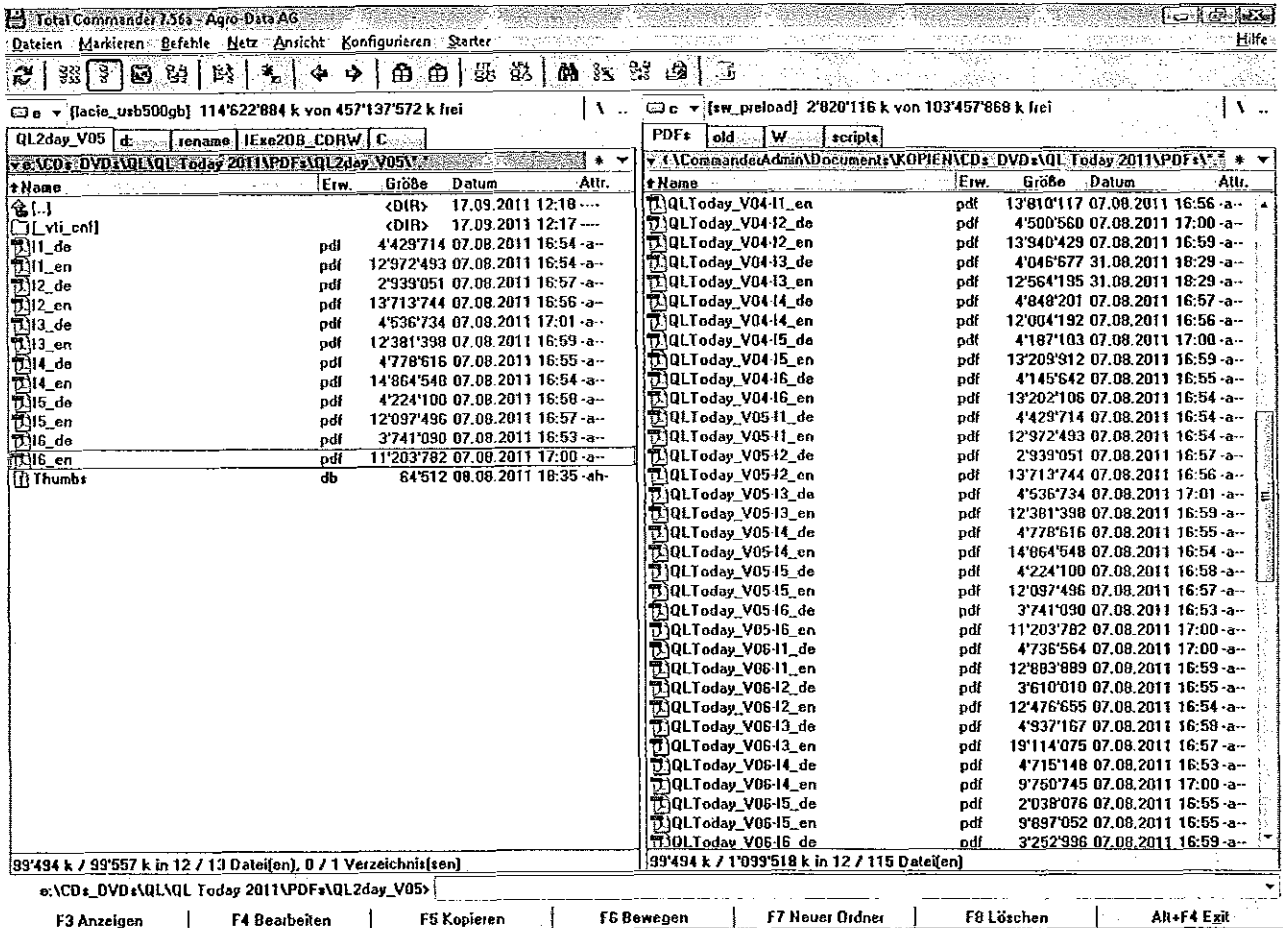
Even though this brought down the total size by only by a few MBytes, it brought down the number of files and folders by more than 50% which is quite important to lighten the work of the file system.



Pic. 1) The unneeded `"_vti_cnf"` (left) and `"Thumbs.db"` (right) files and folders.

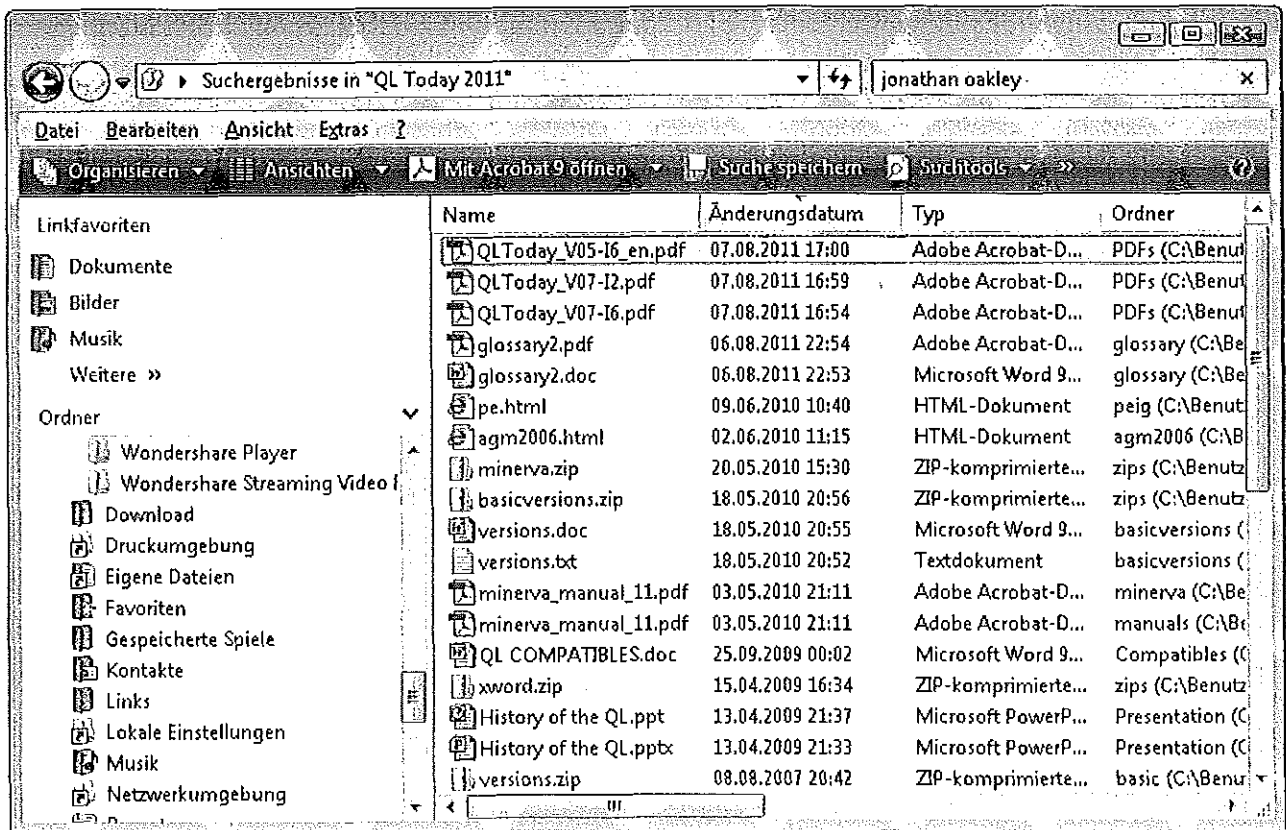
## Rename and reorder the PDFs

While working with the local search engines (software) on my PCs I discovered that at least `"Windows Search"` (standard on Windows Vista and 7, add-on on Windows XP) does find what you are looking for in the scanned magazines. Please remember to make sure that the location where the PDFs are stored is included in the search index. I keep my local copy of the stripped DVD in the `"Documents"` folder of my PCs user account (e.g. `"c:\Users\Urs\Documents\KOPIEN\CDs_DVDs\QL Today 2011"`). So far, so good. But I personally found it an inconvenience that the PDFs have a file/foldername convention (e.g. filename `"l6_en.pdf"` in folder `"\PDFs\QL2day_V05\"`). For my own use I decided to give each magazine a unique and full featured filename (e.g. `"QLToday_V05-l6_en.pdf"`) and place all magazines in the PDFs root folder. As a result of this the two magazine navigation menus `"index english.htm"` and `"index deutsch.htm"` needed to be amended.



Pic. 2) Filename and folder convention pre (left) and post (right) rename and reorder.

One of many benefits of this is that "Windows Search" nicely displays a meaningful filename. I assume the same benefit will be achieved on Mac OS X using the "Spotlight" search engine.



Pic. 3) Benefit of having the new full-featured filename (e.g. "QLToday\_V05-I6\_en.pdf") instead of just "I6\_en.pdf").

## Complete the menu

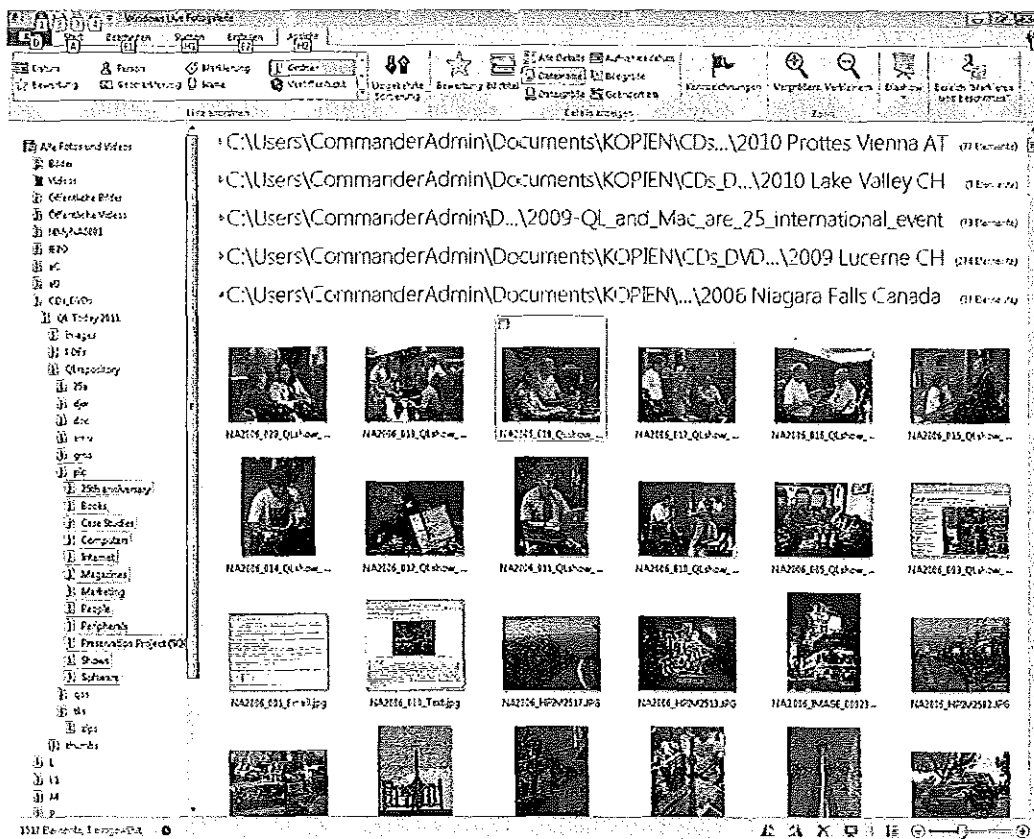
The free demo version of QPC2, more a virtual QL system than just an emulator, is easy to start using the DVDs menu ("index.htm"). What's missing are links in the menu to its folder and to its "README first" document. For my own convenience I've added those two links to the menu.

## Picture galleries easy-peasy<sup>2</sup>

The DVD holds a total of 3833 images, of which 117 are for the menu and magazine navigation. The remaining 3715 images are stored in the various collections in the "\QLrepository\" folder. About 40% (1517 images) are stored in the "QL PICTURE GALLERY". Another 35% (1322 images) are stored in the "QL HISTORY, 25TH ANNIVERSARY edition". The remaining 25% are mainly stored in the "Local (offline) copy of Dilwyn Jones's QL website" (594 images) and the "QL DOCUMENTATION collection" (221 images).

The easiest way to access/handle the images is to use/browse them with picture gallery software like Microsoft's "Windows Live Essentials 2011 Photo Gallery"<sup>3</sup> (W LPG for short) or Google's "Picasa 3"<sup>4</sup>. Both software packages are free downloads at no cost. As long as you've stored the DVD's content to a location known and managed by those software packages (e.g. the Documents folder of your PC's user account) then nothing more needs to be configured. If you like to store the DVD's content in another place (e.g. on your file server), then you must add the network share to your picture gallery software's managed locations. Once this is done, you can use all the features of the software.. I prefer to use Microsoft's "Windows Live Essentials 2011 Photo Gallery". It's easy to use, smart and fast at browsing through your picture collection, good to order pictures and has some nice picture edit features. Users

of Windows XP need to download and use the 2009 version of this package. Some Windows users may prefer using Google's "Picasa 3" or another picture manager of their choice. Mac OS X users may use "iPhoto"<sup>5</sup>. The only features I like more in "Picasa 3" than in W LPG are the "Geo-Tag/World-Map" view (W LPG has no "World-Map" feature at all, it only displays the location name based on the Geo-Tag), its search option and the "Timeline" browser.



Pic. 4) W LPG in action. I like the easy to use and speedy environment of this all in one package which is greatly integrated into the Windows system (no Wonder as it's from Microsoft).

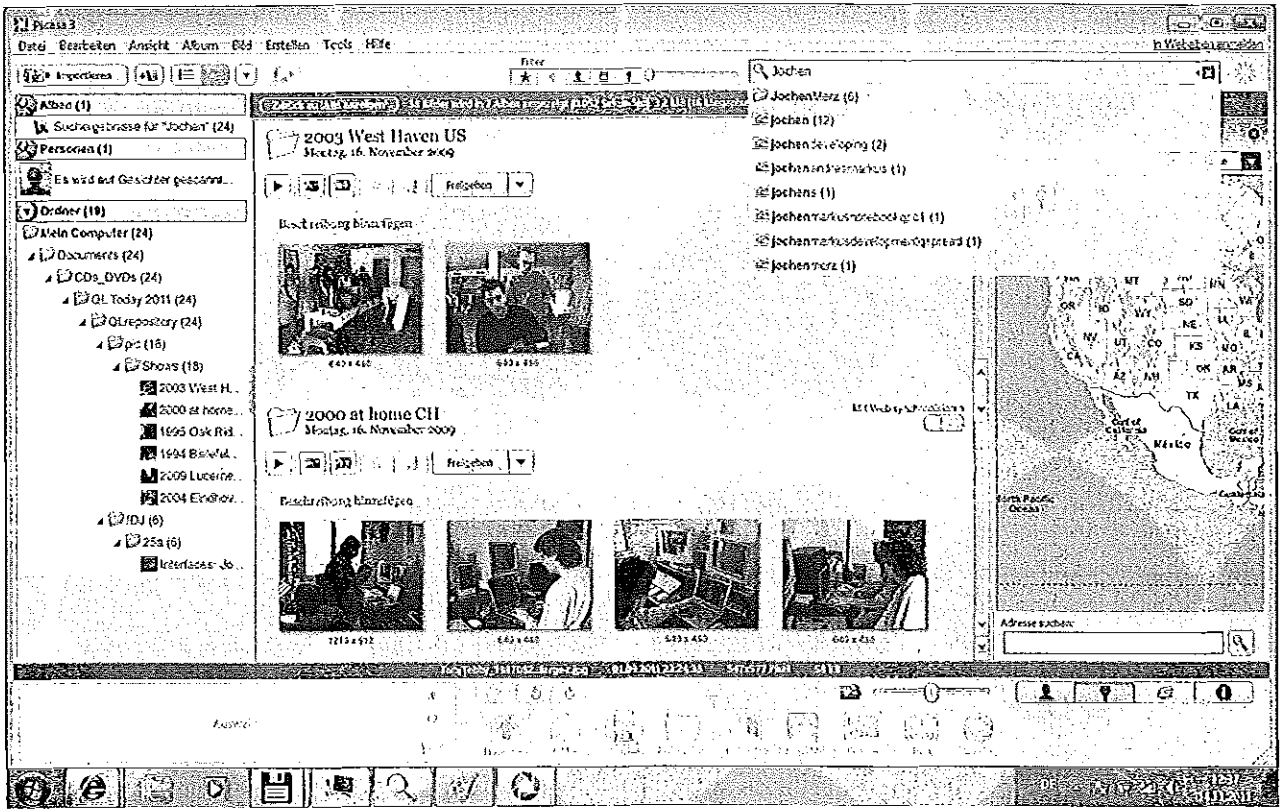
2 <http://www.urbandictionary.com/define.php?term=easy%20peasy>

3 [http://en.wikipedia.org/wiki/Windows\\_Live\\_Essentials](http://en.wikipedia.org/wiki/Windows_Live_Essentials)

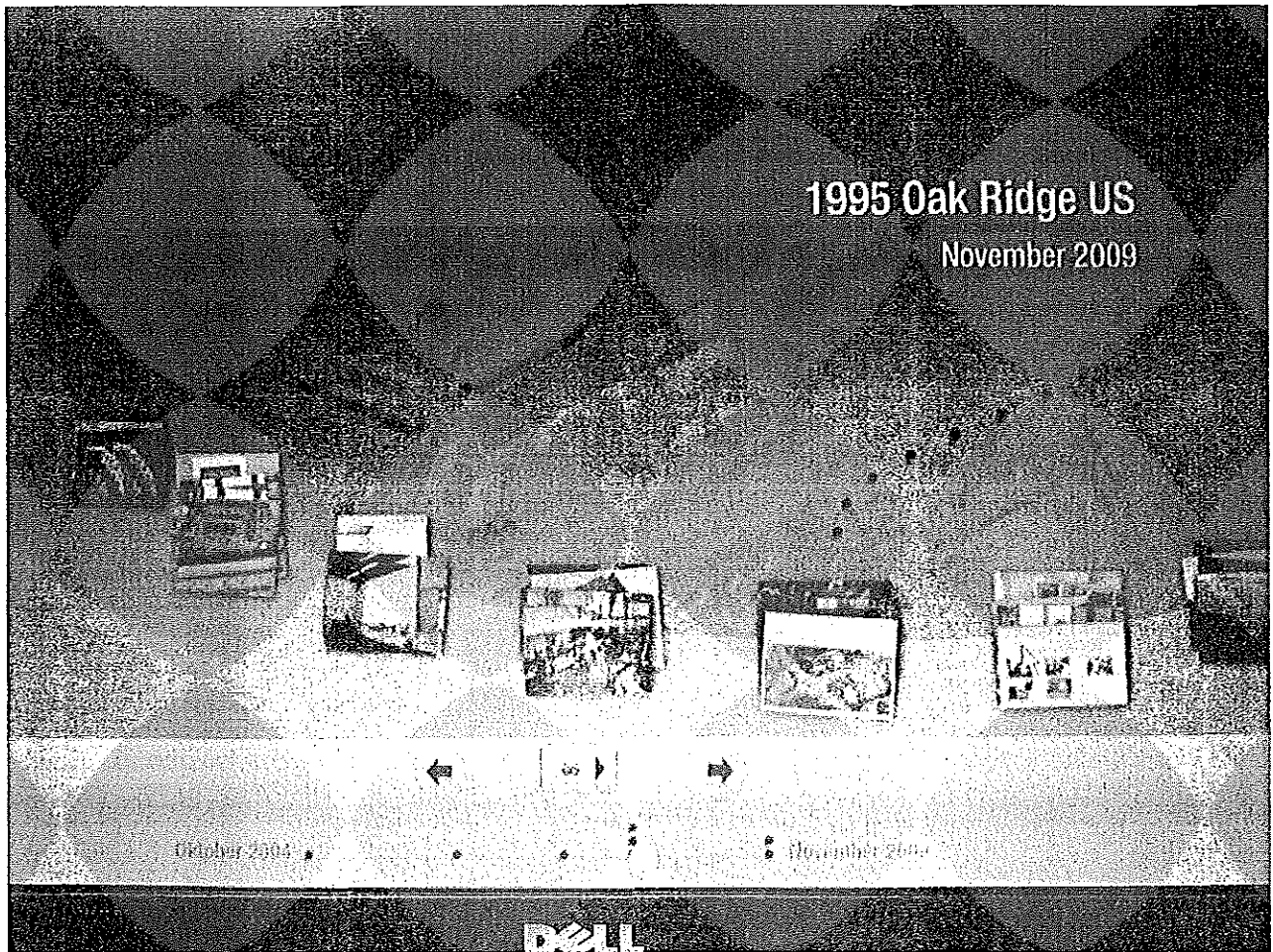
4 <http://en.wikipedia.org/wiki/Picasa>

5 <http://en.wikipedia.org/wiki/Iphoto>





Pic. 5) Picasa is not that convenient/fast at browsing through folders, but has a more accurate search option, the "Geo-Tag/World-Map" view and...



Pic. 6) ... a "Timeline" browser which is quite nice.



## Virtual WINchester Harddrives (QXL.WINs or the like)

There are a total of 13 virtual WINchester hard-drives in QXL format (QXL.WIN for short) on the DVD. 5 of them are located in the root folder of the QL software collections (see earlier chapter). Then there's one called QPCDEMOWIN which is part of the demo version of QPC2. This demo version is stored twice on the DVD. First it is included in the "QL EMULATORS collection" and second in the "QL ON A STICK environment". This is a case where duplicates can make sense. Then there are 6 more \*.WINs which are stored in ZIP-Files. 4 of them are backup copies of the ones mentioned above. One (\\QLrepository\djw\language\gstqc\_qxlwin.zip) holds the complete collection of software written by GST for QDOS. This collection has been released into public domain in summer 2010<sup>6</sup>. The last one (\\QLrepository\emu\archives\petawin.zip) is a rather old environment which was prepared way back in 1996 (15 years ago, on Dec 23rd, the day before Christmas Eve) by Peta Jäger for distribution and use with the demo package of QPC version 1.

Starting QPC2 from the DVDs menu will boot from QPCDEMOWIN.

Starting QPC2 using "qpc2.bat" from the "QL ON A STICK environment" will boot from QXL.WIN of that environment. Users of QPC2 can easily configure to use specific file system containers in QXL.WIN format for use as WIN1\_ to WIN8\_. They can even MOUNT such file system containers at runtime using the SBASIC command WIN\_DRIVE. WIN\_DRIVE even supports relative paths based on where QPC2 has been started of (e.g. WIN\_DRIVE 1, ".\..\gms\qxlwin"). User of Q-emulator can mount QXL.WIN based file system containers on the main screen with a right click to the virtual Microdrive slot.

## Hands on

You can strip-down and pimp your DVD in two ways: manually using the information of this article or using a small Windows-Program called

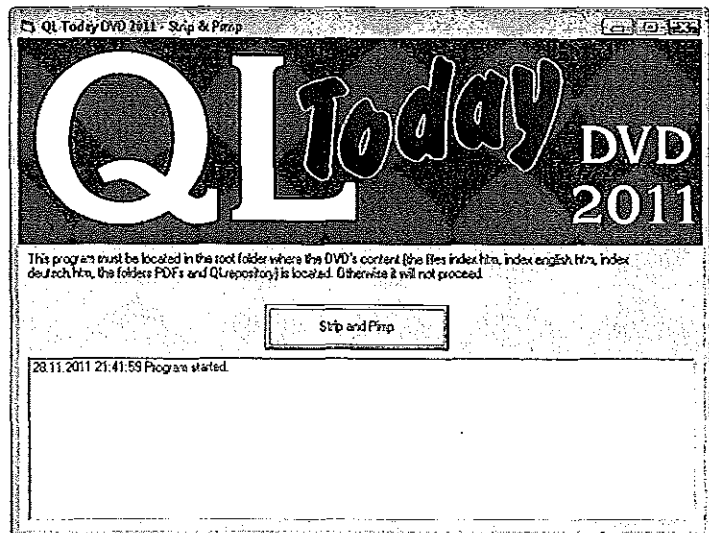
"QLToday\_DVD\_StripAndPimp.exe" which I wrote especially for this purpose.

You can download this program on the following webpage:

<http://tinyurl.com/QLTodayDVD>

Feel free to re-burn a striped-down and pimped DVD for your own use.

Enjoy exploring the DVD!



Pic. 7) GUI of the Windows-Program called "QLToday\_DVD\_StripAndPimp.exe".

6) [http://www.qivs捷豹homepage.bluewin.ch/QL\\_GST\\_SW\\_is\\_PD.html](http://www.qivs捷豹homepage.bluewin.ch/QL_GST_SW_is_PD.html)

## The DVD - Jochen's View

by Jochen Merz

Thanks Urs for adding the final touch to the DVD afterwards. This kind of DVD was the first I ever produced, and you may remember I asked for some help and hints two issues ago. The inbox was rather empty, except from Urs's help.

However, the result was rather good for being the first attempt - and produced in a rush too!

At least we got many appreciative mails from our readers - thanks at least here for the great feedback!

When I thought about adding a DVD I did not

imagine how much time, work and additional costs it produced. The creation of the disk was already time-consuming, and I was not aware of the additional files which Urs mentioned in the previous article. Creating an auto-starting DVD was not as easy as I initially thought - it took some attempts to get it working.

But printing the image onto the surface of all the DVDs (I prefer printable media over labels, as the stickers may come off after some years) took

even longer. And even worse - burning the DVDs took days ... really, many days. I burned all DVDs using the same drive, to ensure, if something goes wrong, I know where the problem came from. The DVDs were only 4x speed, so you can calculate yourself how long it took.

Fortunately, only very few of our subscribers had a problem reading the DVDs ... I was able to read them in all of my various drives (that was something I tested before I burned all of them because I knew there were problems with DVD compatibility, especially with DVD-RWs.

Most of the few problems were fixed by our subscribers by trying a different DVD-drive. Only very few replacements, burned on completely

different media, solved the remaining problems. What a relief!

Still, the whole project required much more expensive envelopes to protect the DVDs, higher postage (and sent from Germany again like last time as I did not travel to Austria) and an enormous amount of my time. I am glad it worked out so well, but it really went over the planned budget. But, as you see, we still do not cut on the number of pages if you send us material!

If you follow Urs' explanation, you will really pimp the DVD ... if you feel you cannot or you do not want to do it yourself, let me know (via email or letter). Maybe, maybe we find a solution, if there is enough interest.

## E.V.A QL Review

by P. Scott

Recently re-released by RWAP Software on floppy disc for £5. EVA is a fast paced single screen arcade shooter written in machine code and originally sold by Westway Ltd in 1985. Very much a tribute to the speccy classic Jetpac. EVA has you dodging fireballs and aliens to collect various parts of robotic defences which must be assembled on a red pad to progress to the next level.

The graphics are fairly plain but functional and crisp in design. On the first level meteorites randomly fizz about crashing into each other as well as the scenery while the second level sees aliens dancing round the screen in patterns.

The actual mechanic of picking up the parts and placing them can be fiddly but with practice you will be swooping about with some flashy graceful moves. It's also possible that the stiff action on my Medic M-stick was making a bit of a job out of it. At one point I was left puzzled when my

little spaceman refused to fire only to find the stick had slid out of the odd RF style socket in the base that handles the fire buttons. The game

also plays fine with the standard cursor keys with not a huge amount of difference between the two.

EVA is an interesting revisit to the arcade space shooter that was commonplace in the early 80s.

Although this type of game has a

questionable depth with it's simple gameplay, it is well worth a quick blast. It's also lovely to see QL games tracked down and re-released which is always worth supporting, especially in the face of high priced Ebay games which might arrive on a faulty microdrive cartridge.

A video clip of EVA together with other clips of QL games is available on my YouTube site at:

<http://www.youtube.com/watch?v=9K0uYrrvN>

QLPeet 12 videos



135 views

Uploaded by QLPeet on 6 Aug 2011

A quick video of the 1985 jetpac clone EVA running on my Sinclair QL.

3 files, 0 dislikes

[Show more](#)

# Programming in Assembler, Part 29

## Application Sub-Window Menus

by Norman Dunbar

### Chapter 1. Application Sub-Window Menus

#### 1. Introduction

At the end of the last issue, I promised to continue looking at Application Sub-Windows by adding Application Sub-Window Menus to them. This issue fulfills that promise.

Effectively, there are two different types of application sub-window menus:

- Static – which are defined in the program source code and never change;
- Dynamic – which may change as the applications runs.

In this issue, we shall look at the Static Application Sub-Window Menus only. In a future article, we shall look at Dynamic Menus.

#### 2. Static Application Sub-Window Menus

Static menus, as I shall call them from this point onwards, are created by the developer as s/he writes the program. As the program runs, static menus do not change – other than setting entries to available or unavailable as required.

We can use SETW to create static menus. All that is required is for the developer to decide on the menu options, the required layout of rows and columns, and what to do when the user clicks on an option – although this latter option is not needed by SETW, only in the application's code.

We need to design a new window using SETW, so proceed to execute the utility and proceed as follows:

1. When prompted for 'name\$' enter *AppMenuTest1*.

Note: I'd like to use the name *AppMenuTest1Win*, but that is too big for SETW. When finished, the file *AppMenuTest1.asm* is easily renamed to *AppMenuTest1Win.asm*.

2. On the 'Alter Text' screen.

- Press N for new, type 'X' (without the quotes) then ENTER.
- Press N for new, type 'Application Menu Test 1' (without the quotes) then ENTER
- Press N for new, type 'One' then enter.
- Repeat for text objects 'Two' through 'Ten'
- Press ESC.

Note: at the end of this stage, you should have 12 text objects defined.

3. On the 'Alter Sprite' screen.

- Press ESC.

4. On the 'Alter Blob' screen.

- Press ESC.

5. On the 'Alter Patt' screen.

- Press ESC.

6. Number of main windows = 1.

7. Number of Loose Items = 1.

8. Number of Information windows = 2.

9. For Information Window 1 of 2, the number of IW Objects = 1.

10. For information windows 2 of 2, the number of IW Objects = 0.10

11. Number of application windows = 1.
12. Application windows menu items = 10.
13. For main window 1:
  - Shadow = 2
  - Border size = 1
  - Border colour = colour\_q1 → black
  - Paper colour = colour\_q1 → white
  - Sprite = arrow
14. Presentation of loose Items:
  - Press N for "system palette defaults"
  - Confirm N when prompted again for defaults
  - Border size = 1
  - Border colour = colour\_q1 → black
  - Unavailable background = colour\_q1 → white
  - Unavailable Ink = colour\_q1 → grey
  - Available background = colour\_q1 → white
  - Available Ink = colour\_q1 → black
  - Selected background = colour\_q1 → green
  - Selected Ink = colour\_q1 → black
15. Loose Item 1:
  - Type = text
  - Object → select the 'X' text object
  - Selection key = ESC
16. Information Window 1:
  - Border size = 0
  - Paper = colour\_q1 → No 92
17. Object 1:
  - Type = text
  - Object → select the 'Application Window Test' text object.
  - Colour = colour\_q1 → black
  - Xcsize = 0
  - Ycsize = 0
18. Information Window 2:
  - Border size = 1
  - Border colour = ql\_colour → black
  - Paper = colour\_q1 → white
19. Application Window 1:
  - Border size = 1
  - Border colour = colour\_q1 → black
  - Paper colour = colour\_q1 → white
  - Sprite = arrow
  - Selection key = TAB
  - Presentation of Menu Items
    - Select N for system palette defaults
    - Select N for defaults, again.
    - Border size = 1.
    - Border colour = ql\_colour → black.
    - Unavailable background = ql\_colour → white.
    - Unavailable ink = ql\_colour → grey.
    - Available background = ql\_colour → white.
    - Available ink = ql\_colour → black.

- Selected background = ql\_colour → green.
- Selected ink = ql\_colour → black.
- Scroll arrow = ql\_colour → white.
- Scroll bar = ql\_colour → black.
- Scroll background = ql\_colour → red.
- When prompted for the ten menu items, select the text items 'One' through 'Ten'. Give each one a selection key of the digit that matches the number described by the text object. For example, 'One' has a key of '1', 'Two' has '2' and so on up to 'Ten' which has selection key '0' (Zero).

20. Main window size: (Use the arrow keys to change the size, press ENTER when correct)

- Width = 220
- Height = 140
- Do you want a variable window = N
- Set the origin to 0,0 (Press ENTER when correct)

21. Loose Item 1: (Toggle hit/position with F2. Press ENTER when correct)

- Hit size = 10 x 10
- Position = 206 x 3

22. Information Window 1: (Toggle size/position with F2. Press ENTER when correct)

- Size = 220 x 16
- Position = 0 x 0
- Object position = 2 x 2

23. Information Window 2: (Toggle size/position with F2. Press ENTER when correct)

- Size = 216 x 14
- Position = 2 x 125

24. Application Window 1: (Toggle size/position with F2. Press ENTER when correct)

- Size = 208 x 104
- Position = 2 x 18

When you have completed this procedure, and SETW has exited, you should save the file ram1\_AppMenuTest1\_asm to a safer place and rename it to AppMenuTest1Win\_asm. The file should look similar to the following, although I have added some extra comments to my copy of the generated code.

### 3. The Generated Code

The file should look similar to the following, although I have added some extra comments to my copy of the generated code.

**Note:** I have removed a few sections of the following file in order to reduce duplication of chunks of code in the magazine. These sections are discussed below.

```
; AppMenuTest1Win_asm.
; Undefined Labels -- need to be defined elsewhere in my own code.
;
;      ahit2_0  -  menu item 0, hit routine.
;      ahit2_1  -  menu item 1, hit routine.
;      ahit2_2  -  menu item 2, hit routine.
;      ahit2_3  -  menu item 3, hit routine.
;      ahit2_4  -  menu item 4, hit routine.
;      ahit2_5  -  menu item 5, hit routine.
;      ahit2_6  -  menu item 6, hit routine.
;      ahit2_7  -  menu item 7, hit routine.
;      ahit2_8  -  menu item 8, hit routine.
;      ahit2_9  -  menu item 9, hit routine.
;      asmnu0   -  User defined setup routine.
```

```

;      adraw0    -   User defined draw routine.
;      ahit0    -   application window 0 hit action routine.
;      afun0_0  -   Loose item 0 hit action routine.

; Labels for External Use
;      mst0     -   menu items status area
;      wst0     -   Window status area
;      wd0      -   Window definition address
;      ww0_0    -   Window default size
;      ww0_1    -   Window button size

SYS_SPR   dc.w      0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
                24,25,26,27,28,29,30,31,32,33,34,35,36, 37

; Text object for "Close" loose item.
txt0      dc.w      txt0_e-2-txt0
          dc.b      "X"
txt0_e    ds.b      0
          ds.w      0

; Text object for caption bar.
txt1      dc.w      txt1_e-2-txt1
          dc.b      "Application Menu Test 1"
txt1_e    ds.b      0
          ds.w      0

;-----
; **** Text objects for the menu items. Removed - see text.
; **** Menu items list. Removed - see text.
; **** Row list. Removed - see text.
; **** Spacing list. Removed - see text.
;-----

; Application window list.
app_list0
          dc.w      appw0-*
          dc.w      0

;-----
; **** Application window 0 definition. Removed - see text.
;-----

; Information Object(s).
pobl0
          dc.w      138          xsize
          dc.w      10          ysize
          dc.w      2           xorg
          dc.w      2           yorg
          dc.b      0           type
          dc.b      0           spar
          dc.l      0           spce
          dc.w      txt1-*      pobj *
          dc.w      -1

; Information Window(s).
infw0
          dc.w      220          xsize
          dc.w      16          ysize
          dc.w      0           xorg
          dc.w      0           yorg
          dc.w      0           flag
          dc.w      0           borw
          dc.w      526         bore
          dc.w      92          papr

```



# Q U A N T A

## Independent QL Users Group

World-wide Membership is by subscription only,  
Offering the following benefits: Bimonthly Magazine - up to 52 pages  
Massive Software Library - All Free! : Free Helpline and Workshops  
Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

1 year Membership Subscription £18 (includes eMag)

UK Postage £2.50 - Europe Postage £10

International Postage £14 - PayPal Surcharge £1

PayPal (see QUANTA Web Site), Cash, Cheques and Postal Orders Accepted

\*Now in our Twenty Eighth Year\*

Further details from the Membership Secretary

Keith Dunbar, 44, Dalton Avenue,  
Stretford, Manchester, M32 9TP (UK).

Tel. 07789 483 800

Email: [membership@quanta.org.uk](mailto:membership@quanta.org.uk)

<http://www.quanta.org.uk>

*Email: [membership@quanta.org.uk](mailto:membership@quanta.org.uk)  
and ask about our special  
3 Year discount*

## Next QUANTA Sponsored Event

Annual General Meeting 2012 And Workshop, Date: Sat/Sun 24th/25th March

Opening on Saturday at 11.00 am (Doors open 10 am for setting up),

Sunday at 10.00 am with the AGM at 2.00 pm in the afternoon.

The Venue will be in Manchester at the

3rd Davyhulme Scout Headquarters, "The Endeavour",

Conway Road, off Lostock Road, Davyhulme,

Manchester. M41 0TF. Near M60 J9.

```

dc.w      pobl0-*      pobl *
dc.w      216          xsize
dc.w      14           ysize
dc.w      2            xorg
dc.w      125          yorg
dc.w      0            flag
dc.w      1            borw
dc.w      0            bore
dc.w      7            papr
dc.w      0            pobl *
dc.w      -1           end

; Loose Item(s).
litm0
dc.w      10,10        xsize, ysize
dc.w      206,3        xorg, yorg
dc.b      0,0          xjst, yjst
dc.b      0,3          type, skey
dc.w      txt0-*       pobj *
dc.w      0            item
dc.w      afun0_0-*    pact *
dc.w      -1           end

litm1
dc.w      16404,12     xsize, ysize
dc.w      0,0          xorg, yorg
dc.b      0,0          xjst, yjst
dc.b      0,0          type, skey
dc.w      0            pobj *
dc.w      0            item
dc.w      0            pact *
dc.w      -1           end

; Window definition
wd0
dc.w      220          xsize
dc.w      140          ysize
dc.w      0            xorg
dc.w      0            yorg
dc.w      258          flag
dc.w      1            borw
dc.w      0            bore
dc.w      7            papr
dc.w      0            sprt *
dc.w      1            curw
dc.w      0            curc
dc.w      7            uback
dc.w      255          uink
dc.w      0            ublob *
dc.w      0            upatt *
dc.w      7            aback
dc.w      0            aink
dc.w      0            ablob *
dc.w      0            apatt *
dc.w      4            sback
dc.w      0            sink
dc.w      0            sblob *
dc.w      0            spatt *
dc.w      0            help
dc.w      220          xsize
dc.w      140          ysize
dc.w      infw0-*      pinfo *
dc.w      litm0-*      plitem *
dc.w      app_list0-*  pappl *
dc.w      16384         xsize
dc.w      12           ysize
dc.w      0            pinfo *

```

```

        dc.w      litm1-*   plitem *
        dc.w      0        pappl *
        dc.w      -1
; Sizes
ww0_0     equ      670
ww0_1     equ      148
; Status Areas:
; Menu item status      area.
mst0      ds.b      10
mst0_e    ds.b      0
          ds.w      0
; Window status area.
wst0      ds.b      65
wst0_e    ds.b      0
          ds.w      0

```

Much of the above is similar to when was discussed in a previous article. If you think back to that article on application sub-windows, we simply used the hit routine to print text all over the application sub-window. That was about as simple as it gets - other than not actually having a hit routine I suppose! Adding a menu to an application sub-window means we have quite a lot more work to do at the coding stage as we have to consider the following:

- Defining the menu objects - these can be text, sprite etc. Every menu item must be defined.
- Defining the menu items list - when we have defined each menu object, we then have to build a list of all the menu items that we wish to include in our final menu.
- Defining the menu row list - when we have the menu items list defined, we amalgamate that list into a menu row list which defines the start and end of each row in the menu.
- Defining the spacing lists - the row list defines the hit size and the spacing for each row and each column in the menu.
- Define the application sub-window - menus need their own section in the application sub-window definition.

So much to do just to show a menu in a window, lets get on and do it.

#### 4. Menu Objects

The first stage is to define the various objects that will be incorporated into the menu. In this example, I have used ten separate text objects (as they are the simplest). You can use any of the various Pointer Environment object types if you wish. The code generated by SETW for these items is as follows.

```

; **** Text objects for the menu items.

txt2      dc.w      txt2_e-2-txt2
          dc.b      "One"
txt2_e    ds.b      0
          ds.w      0

; **** Txt3 through txt10 deleted for brevity.

txt11     dc.w      txt11_e-2-txt11
          dc.b      "Ten"
txt11_e   ds.b      0
          ds.w      0

```

As you can see from the above, I have omitted to show text objects txt3 through txt10 as there is really no need to take up space in the magazine with repetitive data.

The above code simply defines ten separate text objects - 'One', 'Two', ....., 'Nine' and 'Ten' to be used, later, in our static menu.

## 5. Menu Items (and Index) List

The next section of code that I have removed from the main listing above is the menu items list. This is shown below, but please note that once again, I have removed the vast majority of the code for brevity.

```
; **** Menu items list.

meos2      dc.b      1,0          x_justification, y_justification
           dc.b      0,49        Item type, selection key
           dc.w      txt2-*      Pointer to object.
           dc.w      0           Item number. (-1 for indexes)
           dc.w      ahit2_0-*   Pointer to action (hit) routine for this item.
;                                                  Zero for indexes.

; NOTE: Menu items 1 through 8 removed here for brevity.

           dc.b      1,0          x_justification, y_justification
           dc.b      0,48        Item type, selection key
           dc.w      txt11-*     Pointer to object.
           dc.w      9           Item number. (-1 for indexes)
           dc.w      ahit2_9-*   Pointer to action (hit) routine for this item.
;                                                  Zero for indexes.
```

The label meos2 is the start of the menu items list. We told SETW that there would be 10 items, and so, there will be 10 menu items in this list. Each one has the following structure:

Offset	Size	Description
0	1	X (Horizontal) justification. <ul style="list-style-type: none"><li>• -ve = Right justified</li><li>• 0 = Centred</li><li>• +ve = Left Justified</li></ul> The justification value is the number of pixels from the edge of the hit area that the object is to be positioned at.
1	1	Y (Vertical) justification. <ul style="list-style-type: none"><li>• -ve = Bottom justified</li><li>• 0 = Centred</li><li>• +ve = Top Justified</li></ul> The justification value is the number of pixels from the edge of the hit area that the object is to be positioned at.
2	1	Item type: <ul style="list-style-type: none"><li>• 0 - Text object</li><li>• 2 - Sprite object</li><li>• 4 - Blob object</li><li>• 6 - Pattern object</li></ul>
3	1	Selection key, upper cased if necessary.
4	2	Relative pointer to the actual object.
6	2	Item number. If this is an index items list, set to -1 for all items.
8	2	relative pointer to the hit/action routine for this particular item. For index item lists, must be zero.

Each item in a menu items list is 10 bytes in length.

Looking at the first menu object generated by SETW, we see that it is left justified and one pixel from the start of the left end of the hit area. Vertically, it is centred within the hit area. It is a text object and has a selection key of 49, which is the code for a digit '1' on the keyboard. The text object itself is the text 'One' as shown above in the preceding section. The item is numbered zero and the hit routine for this item is defined to be at label ahit2\_0.

So far, so good. We have defined a list of objects and then gathered them into a list of menu items. The menu items list - in this case - is in a contiguous section of memory, it need not be so. The row list defines the menu ordering, that comes next.

The above structure is used to define menu item lists and also index lists. An index is drawn by the WM\_INDEX vector (which also draws pan and scroll bars & arrows - if necessary). Indices are best thought of as the row and column headings - similar to a spreadsheet, for example, when columns have letters and rows have numbers to identify them.

WMAN takes care of aligning the indices with the contents of the static menu.

## 6. Row List

The row list takes the various menu items, defined above, and organises them into rows - surprisingly enough. For every row you wish to have in your menu, you need a single row list entry. As SETW tries to make as few rows and/or columns as it can - it tries to fit as much as possible into a given space - what SETW generates may not be what you want. IN the default case for our SETW session, we have been given two rows and thus, five columns, for our ten menu items.

Our generated row list is as follows:

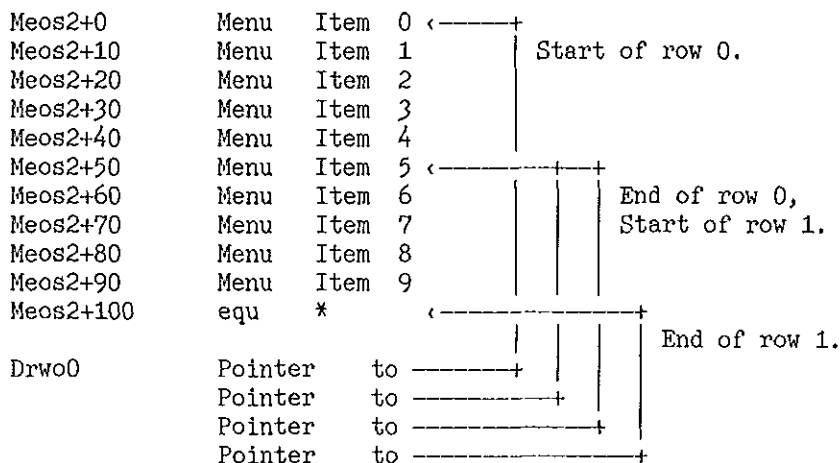
```
; **** Row list.

drow0      dc.w      0+meos2-*           Pointer to row 0 start = item 0.
           dc.w      50+meos2-*        Pointer to row 0 end = item 4.
           dc.w      50+meos2-*        Pointer to row 1 start = item 5.
           dc.w      100+meos2-*       Pointer to row 1 end = item 9.
```

Each row list item contains two pointers, the first is to the start of the first entry in the menu items list entry for this row. The second pointer is to the first byte past the end of the last menu items list entry for this row.

Given the above then, we can see that the first row, starting at label drow0, begins at meos2 (relative to the pointer itself - as usual). Meos2 is a list of 10 sets of 10 byte entries defining all ten items in our menu. The first row ends at meos2 + 50, which happens to be the very first byte of the menu items list for menu item number 5 (ie, the sixth menu item - we count from zero)

The second row list entry starts at meos2 + 50 and ends at meos2 + 100. These pointers are to menu items list item number 5 and at the first byte past the very last menu items list entry. As the following diagram attempts to display:



If the menu items list is a single chunk of memory, then each row start pointer is equal to the previous row end pointer – except for the first row. As in the example above, the end pointer for row 0 is the same address as the start pointer for row 1.

Now that we have our rows defined, we have to set up the spacing lists for each row and each column in the menu.

## 7. Spacing Lists

Each menu item in our static menu has a given hit area and a spacing. The hit area defines where the pointer can be to make the item beneath it the current item, this is normally indicated by a border being drawn around the current item. A HIT or a DO within the hit area, or a press of the selection keystroke while the pointer is within the application sub-window, will cause the appropriate menu item action routine to be executed.

The spacing defines how many pixels across – or down depending on whether this is the column or row spacing list – there are between the start of 'this' menu item and the start of the 'next' one. The spacing must include an allowance for the border to be drawn around the current item.

```
; **** Spacing lists.
```

```
; Spacing list. Defines size of hit area for each COLUMN and spacing  
; between columns. (5 columns.)
```

```
spls0    dc.w    34      Size of hit area for column 0.  
         dc.w    36      Space between this column and the next.  
  
         dc.w    34      Size of hit area for column 1.  
         dc.w    36      Space between this column and the next.  
  
         dc.w    34      Size of hit area for column 2.  
         dc.w    36      Space between this column and the next.  
  
         dc.w    34      Size of hit area for column 3.  
         dc.w    36      Space between this column and the next.  
  
         dc.w    34      Size of hit area for column 4.  
         dc.w    36      Space between this column and the next.
```

```
; Spacing list. Defines size of hit area for each ROW and spacing between rows. (2 rows.)
```

```
spls1    dc.w    10      Size of hit area for row 0.  
         dc.w    12      Space between this row and the next.  
  
         dc.w    10      Size of hit area for row 1.  
         dc.w    12      Space between this row and the next.
```

The first list above, at label spls0 defines the columns in our menu. We already know that SETW has decreed that there shall be five columns and two rows, so the column spacing list has five entries, one for each column. Each entry consists of a pair of words – the first defines the width of the column (or the height of the row) and the second defines the space between this column and the next.

In the example above, we see that SETW has calculated that our widest text object is 5 characters wide – this corresponds to 'Three', 'Seven' and 'Eight' – and has allocated 34 pixels of hit area for each column. The spacing for each columns is set to the (border width \* 2) plus the hit area width. It must be twice the border width as there is a border on each side (or top & bottom).

The spacing list for the rows shows a height of ten pixels for the hit area and taking the border into consideration again, a spacing of 12 pixels between the tops of each row.

## 8. Menu Section of Application Window Definition

The application window definition needs an extra section adding after the normal definition, to cover

the need for a static menu. In addition, two entries in the normal definition part are amended (from what we used for an application sub-window without a menu – see last time) to point to the:

- User defined setup routine, or zero if not required.
- User defined drawing routine, or zero if not required.

The new style application window definition is as follows:

```

; **** Application window 0 definition.

appw0    dc.w        208        Width in pixels (+ scaling)
         dc.w        104       Height in pixels (+ scaling)
         dc.w         2        X origin, relative to 0 in main window
         dc.w        18       Y origin, relative to 0 in main window.
         dc.w        256      Clear flag – bit 7 set = clear window.
         ;              ;      – bit 1 set = disable cursor key movement.
         dc.w         1        Border width
         dc.w         0        Border colour
         dc.w         7        Paper colour
         dc.w         0        Pointer to pointer sprite, or zero for arrow

```

The first part is exactly as we used last time, nothing different to see here.

; Note the following for menus.

```

         dc.w        asmnu0-*   Pointer to user defined setup routine, or zero
         dc.w        adraw0-*   Pointer to user defined drawing routine, or zero
         dc.w        ahit0-*    Pointer to hit routine
         dc.w         0          Pointer to sub-window control routine, or zero
         dc.w         0          Max allowed X control sections (splits)
         dc.w         0          Max allowed Y control sections (splits)
         dc.b         9          Selection key – moves pointer into this window
         dc.b         0          Spare byte – must be zero

```

The first two entries in the above definition are the new ones. These are our pointers to a user defined setup routine and a user defined drawing routine. You will notice that the application window still has its own hit routine, even though it contains a menu and each and every menu item has a dedicated hit routine of its own. Note also, in this small example, that our settings for the pan and scroll sections are all unused. We'll come back to those in a future article.

The user defined setup code would normally consist of a single line as follows:

```
asmnu0    jmp wm_smenu(a2)          ; Vector $08
```

Similarly, the user defined drawing routine need only perform the following tasks:

```

adraw0    jmp wm_index(a2)         ; Vector $34
         bne.s adexit             ; Bale out on errors
         jmp wm_mdraw(a2)        ; Vector $20
adexit    rts

```

The call to `wm_index` is not required unless your menu has been defined to have sections and/or index items. What are index items? Think of a spreadsheet, each row has a number and each column has a letter. These are the index items. Our example is not using index items, however, if it did then we would set them up exactly as per the menu items list, except, for indexes the list entries have no hit routine (set to zero) and the item number is always -1.

**Note:** If the pointer to the user defined drawing routine is zero, then WMAN will still draw the application sub-window's border and, unless the flag is set to say "do not clear", will clear it to the defined paper colour. If you find missing menus in your application sub-windows, check that you have a drawing routine!

Following on from the above, there is a brand new section dedicated to the menu.



; The following section is required when an application sub-window contains a menu.

dc.w	mst0-wst0	Pointer to menu status area. (See text)
dc.w	1	Current Item, border width.
dc.w	0	Current Item, border colour.
dc.w	7	Unavailable background colour.
dc.w	255	Unavailable ink colour.
dc.w	0	Unavailable blob pointer.
dc.w	0	Unavailable pattern pointer.
dc.w	7	Available background colour.
dc.w	0	Available ink colour.
dc.w	0	Available blob pointer.
dc.w	0	Available pattern pointer.
dc.w	4	Selected background colour.
dc.w	0	Selected ink colour.
dc.w	0	Selected blob pointer.
dc.w	0	Selected pattern pointer.
dc.w	5	Number of columns in the menu.
dc.w	2	Number of rows in the menu.
dc.w	0	X offset to start of menu.
dc.w	0	Y offset to start of menu.
dc.w	spls0-*	Pointer to column spacing list.
dc.w	spls1-*	Pointer to row spacing list.
dc.w	0	Pointer to column index list.
dc.w	0	Pointer to row index list.
dc.w	drow0-*	Pointer to menu row list.

The first new entry we need is a pointer to the menu items status area. This has been defined for us, by SETW, at label mst0. There should be a single byte for each menu item. Note however, that we need to have this status area pointer defined as relative to the window status area. Hence the calculation in the above definition.

**Note:** This fact is not very clearly documented in the PE documentation. I had an extended conversation with George on this setting as I had never seen the fact that the menu status area pointer is relative to the window status area - George had a pencilled in note in his copy of the documentation indicating this need. Obviously, I didn't.

Next up, we see the menu attributes - border width and colour, item paper and ink, blobs and patters for unavailable, available and selected items.

After the attributes section, we define the menu itself with details of how many columns there are, how many rows, offsets to the start of the menu and the pointers to the various sections discussed above.

## 9. Application Sub-Window Menu Item Hit Routines

In addition to the application sub-window's own hit routine, as described previously, each and every item in the menu (Static or dynamic) may also have a hit routine. This routine could be a single one for all, or a separate one for each menu item. It depends on how the program is designed.

**Note:** Whenever a program has a static or dynamic menu, there must be a hit routine for the application sub-window containing the menu. The absolute minimum code in the hit routine is as follows:

```
ahit0      jmp wm_hit(a2)          ; Vector $34
```

If you do not have the above code present in the hit routine for the application sub-window, then when you attempt to hit or do a menu item, nothing will work. The above code does not need an RTS.

On entry to a menu item hit routine various registers are set with specific parameters:

Register	Description
D1.L	Virtual column/row for the hit menu item.
D2.W	Item number.
D4.L	An event number. This can only be 0 or pt_do (16).
A0.L	Channel id.
A1.L	Pointer to the menu status area.
A2.L	WMAN vector.
A3.L	Pointer to sub-window definition.
A4.L	Pointer to window working definition.

Registers not mentioned above are free for use as they are not used by the hit routine.

Hit routines should exit with D5 - D7, A0 and A4 preserved to the same value that they had on entry to the routine. D1 - D3, A1 - A3, A5 and A6 are undefined on exit (which means that they don't care what value they have.) D4.B must be either zero or a window event to be set on exit.

D0 should contain zero or an error code and the SR must be set according to the value in D0 on exit.

**Note:** D3, on return from a hit routine, should normally be returned as per its value on entry. It is not used by `wm_rptr` however, it is used by `wm_rptrt` (read pointer with return on timeout) from WMAN 1.5 onwards. `Wm_rptr` ignores the upper word of D3. If your read pointer loop is using the `wm_rptrt` vector instead, and you have changed the value of D3 within the hit code, you must clear the high word on exit.

On exit, if D0 is clear and the status (Z) bit is set, control will return to the `wm_rptr` loop and not to your application's code. To return to your own code, the hit routine needs to set at least one event bit in the event vector which can be done by returning a suitable value in D4.B on exit.

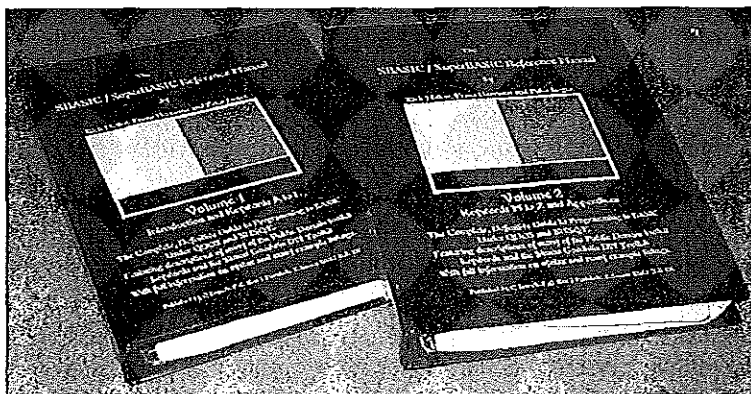
If an error is detected within the hit code, then it should exit with the appropriate error code in D0 and the status register set accordingly.

## 10. End Of Chapter 29

So, that's the end of this exciting instalment. We have designed a window and looked deep into the structures involved in defining a static menu. Next time, we'll add some code and play around. We might even see if it's possible to take the design from SETW and massage it to suit our own design considerations. See you then.

# SBASIC/SuperBASIC Reference Manual

by Dilwyn Jones



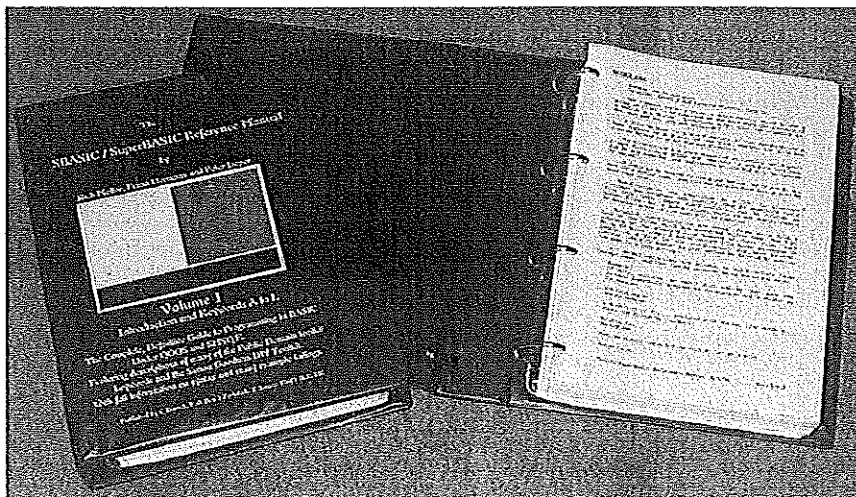
Back in the 1990s, I bought a copy of the original SBASIC/SuperBASIC Reference Guide from Q-Branch. Originally the work of Rich Mellor, Franz Hermann and Peter Jäger, this was a massive work consisting of two A4 ring binders crammed full with hundreds of pages, over a thousand pages in total between the two volumes.

Each volume was roughly the size of the original QL manual, so although it was crammed full of valuable information, it did take up a lot of space on your bookshelf and on your desk when you needed to use it.

The guide was originally written in the early 1990s and sought to build on the original keywords and concepts sections of the QL manual, and to go further by adding details of extensions to QL BASIC provided by Minerva, Toolkit 2, SMSQ/E, Thor and most freely available toolkits. And it has been updated many times since.

A huge section of the publication is devoted to lists of just about every QL BASIC keyword and extensions you are ever likely to meet, along with detailed explanations. As such it provides one of the ultimate guides to QL BASIC available. But in addition it also covers topics such as QL emulators, expansion boards, the extended display systems, new colours, multiple BASICs (MultiBASIC on Minerva and SBASICs on SMQ/E), error messages, international QL version features (messages, keyboard layouts, etc), device drivers, mouse systems and network systems.

Since then, the books have been released as a CD-ROM by RWA Software. This has involved converting the text of the book into PDF files, which can be read using PDF file readers available on most computer systems. This makes it very useful for the large number of QL users whose 'QL' is mainly an emulator running on systems such as Apple Mac, Windows and Linux, for example.



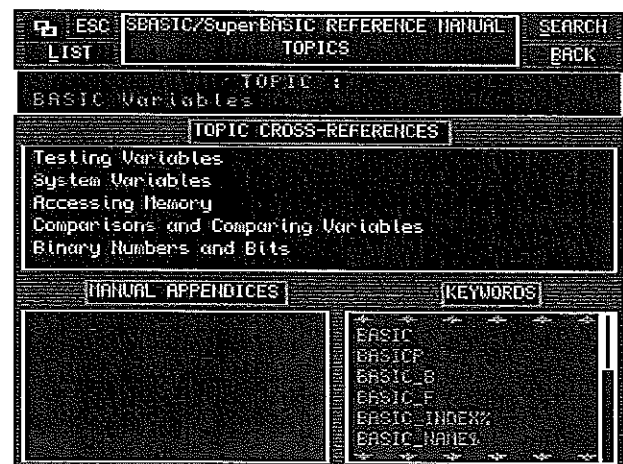
The original Reference Guide as two substantial books

The content of the CD-ROM is organised largely along the lines of the original paper version, but split into manageable size PDF files. The largest section in the manual is of course the keyword

descriptions, and these are stored in their own directory on the CD, split into seven PDF files in sizes of up to a megabyte each, organised as an A-Z list of keywords covering just about every extension to QL BASIC you can imagine.

Another directory has a list of the various appendices from the original guide. Whereas the keyword guide is extensive and full of information, these appendices branch out into all sorts of subjects as diverse as Mathematics and Mouse Drivers – 18 PDF files in all.

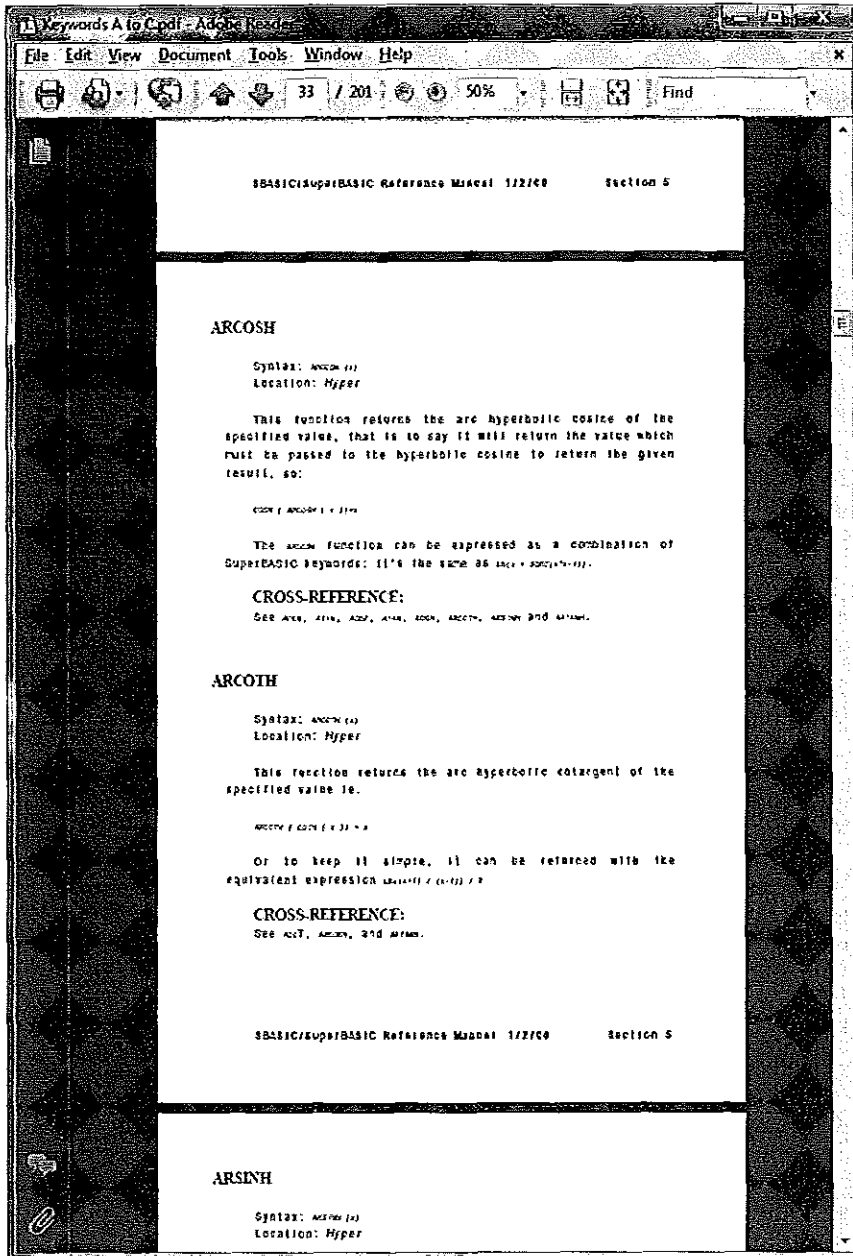
Another directory, called simply 'examples', contains over 200 example listings from the book. Mainly fairly short BASIC listings, they are useful in seeing and understanding how to use many of the keywords, without having to type in the listings from a book.



Q-Index screen

The Q-Index directory contains a QL program of that name, which enables you to find the relevant parts of the manual quickly and easily. Using this program you can enter a word you are interested in and it will list all the topics which refer to this word. You can then select the topic you want to look at to help you work out where to look. If you have the Q-Help program from RWA Software (£10), Q-Index can be configured to make Q-Help available to Q-Index, thus giving you access to online help on a keyword's syntax.

The Toolkits and Sources directories contain an extensive range of freely available toolkits for the QL which are covered in the main guide – large and small keywords, some of the toolkits only contain one or two keywords. The assembler



A page or two from the keyword guide part of the guide

source files are there for many of them, allowing you to study the code for many of the subjects and keywords included. Not always the latest version but at least once you decide you need the latest version of a fast-developing system such as Turbo Toolkit you can search the QL websites out there for the latest version.

For anyone interested in BASIC programming on the QL, be it in either SuperBASIC or SBASIC, this is a pretty useful if not essential guide to BASIC on the QL. It goes into more detail about the keywords than just about any other such guide you can get. And importantly it goes into some detail about programming techniques too,

with many and varied useful examples. The big plus of having it available on CD as a set of PDF files is that it's searchable. Load the relevant PDF into the free Adobe PDF reader program (available from [www.adobe.com](http://www.adobe.com)) and you can simply type in the keyword name and the program will find the details (and any other references) for you.

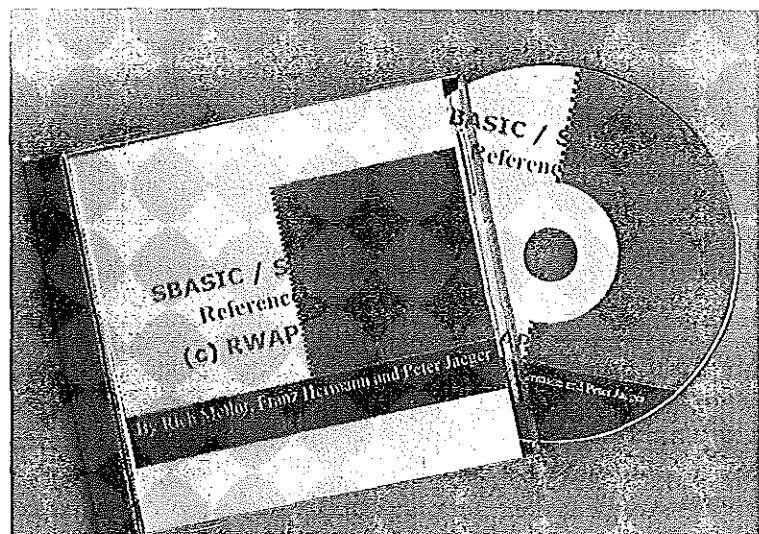
My only real criticism of the SBASIC/SuperBASIC Reference Manual is that it is somewhat undersold and rarely mentioned in adverts. Available to purchase at £20.00 plus postage from

[www.SellMyRetro.com](http://www.SellMyRetro.com) (in the Sinclair QL section), it is one of those major works for the QL which sadly does not receive the recognition that an outstanding work of this nature deserves. You can get some further details from Rich Mellor's website at <http://www.rwapsoftware.co.uk/programming.html>

and a list of prices etc. at <http://www.rwapsoftware.co.uk/prices.html>

Verdict: highly recommended.

Price: £20 plus postage from [SellMyRetro.com](http://www.SellMyRetro.com) or direct from RWAP Software.



The Reference Guide on CD from RWAP Software

# Last-Minute News

Shortly before QL Today went to press three pieces of late news came in.

## Mission Accomplished

Adrian Ives reports the first successful use of the Q-BUS using the ROM port. On November 26, 2011 he wrote on his blog:

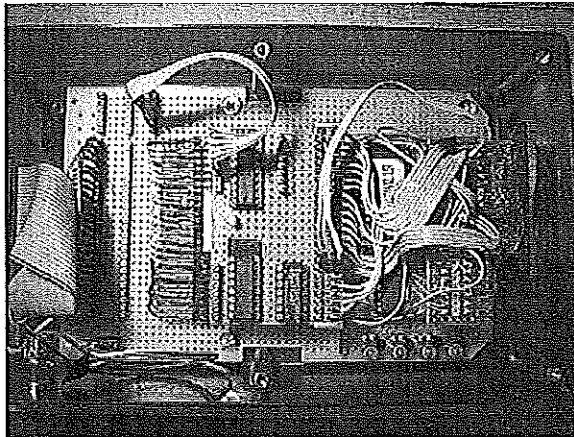
*"At 07:25 GMT this morning, the new USBWiz over Q-BUS driver successfully mounted a QL-format SD card originally prepared on a Ser-USB; the first time that a USBWiz module has been accessed here without using a serial port connection!*

*This may not be the end, or even the beginning of the end, but it is finally the end of the beginning!*

*Work will continue on testing and optimising the driver, but also on reworking the Q-BUS design to use more modern components to make it cheaper and simpler to produce in quantity."*

Adrian asks people who are interested in buying the Q-BUS to contact him so he can ascertain if there is a sufficient demand for a commercial product.

<http://www.memorylanecomputing.com/blog/?p=103>



## Venue Problems

As we go to press QL Today has heard that the Quanta AGM will be held in Manchester for the second year running. The location of Quanta AGMs is becoming a serious problem for the committee.

Traditionally Quanta AGMs were held in alternate years in the north and south of the UK. It was also a firm tradition that the organisation of the AGM show was the responsibility of a local subgroup. Currently there are only two subgroups with the resources to organise a workshop, London and Manchester. The Quanta committee regard the London premises as being unsuitable for a workshop and strenuous efforts by the London subgroup some years ago to find an alternative venue were unfruitful.

In 2009 Quanta tried a central location in Birmingham, but attendance was low.

The 2012 AGM has some important items to consider including the election of a new chairman, the effect of the subscription rise and a revised constitution. In the view of the committee Manchester offers the greatest certainty of a quorate attendance.

## Software Update

George Gwillt writes:

*"SETW, which produces window definitions for TurboPTR, C and EasyPEasy has been updated to version 707. This corrects a fault in the assembler output when blobs or patterns or non-standard sprites are called for. Output for TurboPTR and C should not be affected.*

*(Non-standard sprites are those not supplied with EasyPEasy.)"*

The program is now available on

<http://gwilltprogs.info/>

# Small Ads

Small ads again, and again from Peter!

Maybe most of you forgot about it, but we still offer free private small ads to our subscribers. If you search for something, or you would like to sell or offer something, just send us a letter or an email with the text.

It should, of course, be QL-related, somehow...

## WANTED

Does anyone have the following games:

Fleet Tactical Command by Di-Ren

Gumshoe Logic by Megacycal

Top Team by Arundel Software

Will happily pay for the game please contact

Peter Scott at [peetvanpeebles@yahoo.co.uk](mailto:peetvanpeebles@yahoo.co.uk)



# JOCHEN MERZ SOFTWARE

Kaiser-Wilhelm-Str. 302  
47169 Duisburg, Germany

Fax +49 203 502012  
EMail: SMSQ@J-M-S.com

## XMAS OFFER - UNTIL JAN 2012! 10% DISCOUNT ON ALL PRODUCTS LISTED HERE!

<b>QPC2 Version 3 + SMSQ/E Software QL-Emulator for PC's</b> .....		EUR 59,90
<b>QPC2 Version 3 - Upgrade from QPC2 Version 2</b> .....		EUR 19,90
<b>QPC2 Version 3 - Upgrade from QPC2 Version 1</b> .....		EUR 39,90
<b>QPC Print</b> - printer emulation driver for QPC .....		EUR 39,90
<b>BUNDLE: QPC2 and QPCPrint</b> .....	<b>ONLY</b>	EUR 79,90
<b>Agenda</b> Agenda program for WMAN and ProWess .....	[V1.09]	EUR 14,90
<b>Success</b> Database front-end for WMAN .....	[V2.05]	EUR 19,90
<b>QD2003</b> Pointer-Environment-Editor .....	[VB.01]	EUR 29,90
<b>QD2003</b> Upgrade from Version 9 and older .....	[VB.01]	EUR 14,90
<b>QMAKE</b> Pointer-driven MAKE for GST/Quanta Assembler .....	[V4.31]	EUR 14,90
<b>BASIC Linker</b> .....	[V1.21]	EUR 14,90
<b>WINED</b> Floppy/Harddisk Sector- & File-Editor .....	[V1.26]	EUR 14,90
<b>FiFi II</b> File-Finder - Extremely useful! .....	[V4.31]	EUR 14,90
<b>FiFi II</b> Upgrade from FiFi Version 3 or older .....	[V4.31]	EUR 9,90
<b>EPROM Manager</b> .....	[V3.02]	EUR 14,90
<b>QSpread2003</b> Spreadsheet Program .....	[V4.04]	EUR 29,90
<b>QSpread2003</b> Upgrade from Version 3 and older .....	[V4.04]	EUR 14,90
<b>QPAC I</b> Utility programs .....	[V1.11]	EUR 19,90
<b>QPAC II</b> Files, Jobs & other Things .....	[V1.45]	EUR 29,90
<b>QTYP II</b> Spell checker .....	[V2.17]	EUR 19,90
<b>QPTR</b> Pointer Toolkit .....	[V0.30]	EUR 29,90
<b>DISA</b> Interactive Disassembler .....	[V3.04]	EUR 29,90
<b>CueShell</b> .....	[V2.14]	EUR 29,90
<b>CueShell for QPC</b> .....	[V2.14]	EUR 14,90
<b>SER</b> Mouse software mouse driver for serial mice .....		EUR 10,00
<b>EasyPTR Version 4</b> .....	[V4]	EUR 59,90
<b>EasyPTR Version 4</b> - Upgrade from earlier versions .....	[V4]	EUR 39,90
<b>QDT</b> - QL Desktop program .....		EUR 59,90
<b>QMENU Version 8</b> - with new, printed Manual .....	[V8.02]	EUR 24,90
<b>QMENU Version 8</b> - Update from earlier Versions, also with printed manual .....		EUR 17,90
<b>QMENU Version 8</b> - New/Update for QL Today subscribers, with prt'd manual	<b>ONLY</b>	EUR 14,90

**Please add EUR 4,90 for postage to all destinations - Germany, Europe, Worldwide!**

If you place your order via website SMSQ@J-M-S.com, the full price will be shown  
but the 10% will be taken off manually by me after you placed your order!

### We accept VISA, MasterCard & Diners Club online and offline! Details for money transfers:

- Deutschland: Jochen Merz, Account 493 50 431, Postbank Essen, BLZ 360 100 43
- Österreich: Jochen Merz, Account 85055317, PSK Wien, BLZ 60000
- Switzerland: Jochen Merz, Account 60-690080-4, PostFinance, Clearing-Nr. 09000
- The Netherlands: Jochen Merz, Gironummer 3258439, Postbank NL Amsterdam
- and from all other countries in EUR with IBAN and BIC to account  
Jochen Merz, Deutsche Postbank AG, IBAN: DE21 3601 0043 0611 1004 37 / BIC: PBNKDEFF 360
- UK customers can pay in £ (convert EUR prices above to £ by multiplying with 0.90) to  
Jochen Merz, Account 83795395, Citibank UK, Sort code 30-00-45  
or send cheques in £ - no fee for UK sterling cheques!
- US customers can pay in US\$ (convert EUR prices above to US\$  
by multiplying with 1.39) - no fee for US cheques in US\$!
- If you wish to pay via paypal, send money to Paypal@J-M-S.com

*Cheques payable to Jochen Merz only!*  
Price list valid until 31st of Jan. 2012

# The QL Show Agenda

## Vienna 7th to 10th of June 2012

The meeting at Prottes (near Vienna) is fixed, according to organizer Gerhard Plavec.

The time to reserve in your agenda is the 7th to 10th of June 2012. It is a "long" weekend, like it was last year: Thursday is a bank-holiday (at least in Germany and Austria), so it should make the visit easier.

The schedule for the days is similar to 2010, but other sightseeing highlights are planned:

**Thursday** - arrival, meeting at the Donau-Island in Vienna (nicel!) and a visit of the Tramway museum in Vienna.

**Friday** - visit of the museum village in Niedersulz.

**Saturday** - the main day, meeting at Prottes (like last year) ... let's hope the weather will again be so good!

**Sunday** - last day of the meeting, no special plans yet ...

We hope to have more details in the next issue - remembering how nice the last meeting was, you should look forward to it!

Gerhard will have more details on his homepage by the end of the year:  
[www.kuel.org](http://www.kuel.org)

## QUANTA AGM March 2012

The next QUANTA AGM will be held on the weekend of 24th and 25th of March - more details in the QUANTA ad on page 33 in this issue!

If you have not been to a QUANTA event: The workshop is open for non-QUANTA members too.

You can also visit the QUANTA website:  
[www.quanta.org.uk](http://www.quanta.org.uk)



**MERRY CHRISTMAS  
AND ALL THE BEST  
FOR 2012  
TO OUR READERS  
FROM THE  
QL TODAY TEAM!**